UNIVERSITY OF SOUTHAMPTON

Computer Simulation of Conformational Change in Biological Molecules

Stephen Christopher Phillips

A thesis submitted for the qualification of
Doctor of Philosophy at the University of Southampton

Department of Chemistry

September 2001

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF SCIENCE

CHEMISTRY

Doctor of Philosophy

COMPUTER SIMULATION OF CONFORMATIONAL CHANGE IN BIOLOGICAL MOLECULES

by Stephen Christopher Phillips

A new method for modifying the course of a molecular dynamics computer simulation is presented. Digitally Filtered Molecular Dynamics (DFMD) applies the well-established theory of digital filters to molecular dynamics simulations, enabling atomic motion to be enhanced or suppressed in a selective manner solely on the basis of frequency. The basic theory of digital filters and its application to molecular dynamics simulations is presented, together with the application of DFMD to the simple systems of single molecules of water and butane. The extension of the basic theory to the condensed phase is then described followed by its application to liquid phase butane and the Syrian hamster prion protein. The high degree of selectivity and control offered by DFMD, and its ability to enhance the rate of conformational change in butane and in the prion protein, is demonstrated.

The DFMD method is then modified and extended to become Reversible Digitally Filtered Molecular Dynamics (RDFMD). The RDFMD method improves the degree of control possible over that of DFMD. RDFMD is applied to gas-phase pentane, alanine dipeptide, solvated alanine dipeptide and the pentapeptide YPGDV. In all four systems, RDFMD was able to enhance the rate of conformational change via reasonable transition paths.

Finally, the new method of the Hilbert-Huang Transform (HHT) is described and applied to the analysis of conformational transitions. The HHT is shown to provide clear indications of the changes in energy and frequency during conformational transitions.

Contents

Appendices

List of Figures

List of Tables

Acknowledgements

Firstly I would like to thank my brother, Matthew, for writing most of the TEX code used to typeset this manuscript, for debugging my own attempts and for providing a personal, next-day, inter-library loan service. Without his help I would be using Microsoft Word.

Several members of the Essex group deserve special mention: Oz for being calm, Lewis, Ian and Rich H. for helping me in the early years, David for tediously reformatting lots of figures, Rob for working all hours to get the HHT programs working, and not least Richard Taylor for constant companionship throughout the PhD (till he finished before I did and got a job).

Glaxo SmithKline and particularly Colin Edge have my thanks for sponsoring the work, having faith in the project and being welcoming and hospitable during the time I worked in Harlow.

Jon Essex has been tirelessly enthusiastic about the project, and for that and his friendship I thank him.

Finally, thank you to Jo for looking after the children, the house and me during the last manic weeks of writing the thesis, uncomplaining even though pregnant with our baby girl.

This thesis is dedicated to Jo, Lewis, Tamsin, Felix and ...?

1: Introduction

Conformational change occurring in a protein can be a key step in its biochemical function. The motion can vary from the subtle rearrangements of flexible loops through to large domain rearrangements as observed in some allosteric effects. There are many cases where knowledge of conformational change would aid in designing therapeutic drugs.

Computer simulation by molecular dynamics has the potential for studying in detail the mechanisms of protein conformational change. Unfortunately, the time-scale of many conformational change events is of the order of milli-seconds whereas modern protein molecular dynamics simulations are performed at the nanosecond time-scale. The importance of sampling conformational change is highlighted by the number of algorithms that have been proposed in this field. Many algorithms are limited in that they only produce a transition path between a known start and end point. This is useful in some cases, but in the majority of cases there is no known end-point to target. In this thesis, two new methods for enhancing conformational change requiring no knowledge of the final conformation are presented, along with the first application of a new signal analysis algorithm to molecular dynamics.

The first method, named Digitally Filtered Molecular Dynamics (DFMD), applies

the well-established theory of digital filters to molecular dynamics simulations. Digital filters may be designed to enhance or suppress chosen frequency bands, and it is in this fashion that they are applied to the velocities of an evolving molecular dynamics simulation. By modifying the velocities, selected vibrations may be enhanced or suppressed. In particular, conformational transitions may be induced by enhancing the low frequency motion of dihedral angle vibrations. The excellent selectivity and control of DFMD will be demonstrated and the method will be applied to water, butane and conformational change in the prion protein.

The second method, Reversible Digitally Filtered Molecular Dynamics (RDFMD), extends DFMD to provide more control over the filter application. RDFMD will be shown to be particularly useful in condensed phase systems where energy dissipation is rapid. The method will be applied to pentane, alanine dipeptide and the pentapeptide YPGDV.

In this thesis, the majority of the signal analysis has used Fourier transform methods. Fourier transforms are limited in that they cannot provide time-localised information. The new Hilbert-Huang Transform (HHT) method is ideally suited to the study of the changes in frequency and energy with time during conformational transitions. In particular, a comparison of spontaneous conformational change and RDFMD-induced conformational change validates the decision to enhance the energy of low-frequency modes.

2: Background

2.1: Computer Simulation

Computer simulations are now an invaluable tool in many areas of chemistry. A computer simulation is based on a model derived from theory, and the results from a simulation may be examined to test the theory, and compared with experimental results to test the model. A good model may be used to give insight into the chemical process being simulated by providing detail (in terms of both length- and time-scales) of atomic interactions that would be difficult or impossible to probe with experimental analysis techniques. Also, simulations can just as easily be carried out at any temperature or pressure, including extreme conditions that would be difficult to achieve in the laboratory.

The scale (in terms of length and size) of any computer simulation is limited by the power of the computer and the degree of approximation in the simulation model. A quantum mechanical calculation that explicitly includes a treatment of the electrons, can only be performed on small molecules. By applying the Born-Oppenheimer approximation, an atomistic model may be conceived, where the electronic motion is ignored

Table 2.1. The different scales of simulation techniques.

Technique	Example	Length Scale	Time Scale
Electronic	Wave functions	Bohr radius	Femtoseconds
Atomistic	Molecular interactions	Nanometres	Picoseconds
Mesoscale	Polymers	Microns	Microseconds
Continuum	Fluid flow	Centimetre	Seconds

and only the nuclear coordinates and momenta are considered. A large simulation at this level of approximation would be that of a big protein system. By taking groups of adjacent atoms (such as each peptide in a protein) and considering the groups to be single entities, a mesoscale simulation is possible. Micro-second simulations can be performed at this level of approximation. Ultimately, the system can be considered to be a continuum where all knowledge of even molecular positions is lost. Bulk properties such as fluid flow may be examined at this scale. Table 2.1 summarises the different scales. As the level of approximation is increased, so the level of detail available for analysis decreases. The work presented in this thesis develops methods for examining the conformational change in biological molecules. A detailed picture of the atomic positions is therefore required and so atomistic models are used.

2.1.1: Molecular Mechanics Force Fields

Simulations performed at the atomistic/molecular scale are generally termed molecular mechanics simulations. From the positions and momenta of the atoms, a molecular mechanics force field is used to calculate the approximate energy and forces of a system. Commonly, the force field will consist of terms to take into account the bond lengths, valence angles, torsions, and non-bonded interactions. Widely used force fields, particularly in the area of protein modeling, include OPLS-UA,[1] OPLS-AA,[2] AMBER,[3] GROMOS,[4] and CHARMM.[5] The OPLS-UA force field has been used in the majority of this work, primarily because it is implemented in the programs BOSS,[6] and MCPRO.[7]

The OPLS-UA force field is a united atom (or "UA") force field. This means that hydrogen atoms bonded to carbon atoms are not modelled explicitly, rather the mass of the hydrogen atom is added to the carbon atom to form a "united atom". OPLS is an acronym for Optimised Potentials for Liquid Simulations. The force field was parametrized to reproduce experimental measurements such as the energies and densities

of liquids. The functional form of the total potential energy, U_{total}, in the force field is as follows:

$$U_{\text{total}} = U_{\text{bond}} + U_{\text{angle}} + U_{\text{dihedral}} + U_{\text{non-bonded}} \tag{2.1}$$

The bond and valence angle terms are:

$$\begin{aligned} U_{\text{bond}} &= \sum_{\text{bonds}} K_b (r - r_{\text{eq}})^2 \\ U_{\text{angle}} &= \sum_{\text{angles}} K_\theta (\theta - \theta_{\text{eq}})^2 \end{aligned} \tag{2.2}$$

where the force constants, K_b and K_θ, and the equilibrium values, r_{eq} and θ_{eq} have been parametrized for each type of bond. The dihedral term is represented by a Fourier series:

$$U_{\text{dihedral}} = \sum_{\text{dihedrals}} \left\{ \frac{V_1}{2} \left[1 + \cos(\phi + \delta_1)\right] + \frac{V_2}{2} \left[1 - \cos(2\phi + \delta_2)\right] + \frac{V_3}{2} \left[1 + \cos(3\phi + \delta_3)\right] \right\} \tag{2.3}$$

where ϕ is the angle, V_i are the Fourier coefficients and δ_i are the phase angles for a particular dihedral type. Finally, the non-bonded energy is composed of an electrostatic and a Lennard-Jones term:

$$U_{\text{non-bonded}} = \sum_i \sum_{j>i} \left\{ \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} + 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right] \right\} \tag{2.4}$$

where the sum is over all atom pairs. The q_i are the partial atomic charges, ϵ_{ij} and σ_{ij} are the Lennard-Jones well-depth energy and collision-diameter parameters, and r_{ij} is the inter-atomic distance. In the OPLS-UA force field, the non-bonded term is not calculated for atoms separated by fewer than three bonds and is scaled by 0.5 for atoms that are separated by exactly three bonds (so called "1–4 interactions").

2.1.2: Molecular Dynamics

Molecular dynamics (MD) aims to simulate the real-world dynamics of a system by the solution of Newton's second law of motion, $F = ma$. The force on a particle is calculated from the position derivative of its potential energy, the mass is known and thus the acceleration can be found. If a simulation true to this law is continued for a

long enough time, then the time average of a bulk property measured throughout the simulation will become a good estimate of its true experimental value. To solve $F = ma$ for every particle in the system simultaneously, a finite difference method is generally used.

The Verlet method is a common finite difference method (or "integrator"). The Verlet algorithm has two variants, known as the "Leap-frog Verlet" and the "Velocity Verlet". The majority of the work presented in this thesis uses the leap-frog Verlet integrator:

$$\begin{aligned} \mathbf{r}(t + \delta t) &= \mathbf{r}(t) + \delta t \mathbf{v}(t + \tfrac{1}{2}\delta t) \\ \mathbf{v}(t + \tfrac{1}{2}\delta t) &= \mathbf{v}(t - \tfrac{1}{2}\delta t) + \delta t \mathbf{a}(t) \end{aligned} \tag{2.5}$$

The origin of its name is apparent from its form. The stored quantities are the positions, $\mathbf{r}$, and accelerations, $\mathbf{a}$, at the current time-step, and the velocities, $\mathbf{v}$, half a time-step behind. To calculate the atomic positions at time $t + \delta t$, the velocities at time-step $t + \frac{1}{2}\delta t$ must first be calculated. The velocities "leap-frog" in time over the coordinates and accelerations.

A requirement of a molecular dynamics integrator is that it produces a system with a constant energy. A finite difference integrator will only give a constant energy if a sufficiently small time-step is used. During each small jump forward in time, the integrator's stored quantities are assumed to be conserved. If the jump is too great then this approximation is invalid and the energy will not be constant. A basic Verlet integrator can be used successfully with a time-step of 1 fs. Many of the simulations presented in this work make use of the SHAKE algorithm[8]. This algorithm is used to constrain all the bond lengths to their equilibrium values, and operates via an iterative adjustment process. By constraining the bond lengths, the frequency of the fastest motion is reduced and the integrator can operate successfully with an increased time-step of approximately 2 fs. The overhead associated with the SHAKE algorithm is small compared to the cost of calculating the atomic forces, and so doubling the time-step in this way doubles the amount of simulation time that can be calculated in a given wall-clock time.

2.1.3: Monte Carlo

The other major molecular mechanics simulation method is the Monte Carlo (MC) method. In this work, Metropolis MC simulations have been used for the initial equilibration of a solute in solvent. Like MD, MC produces an ensemble of system states from which a bulk physical property can be calculated. In MD a time average is used but in MC there is no concept of time. A step in a Metropolis MC simulation is the random perturbation of one of the degrees of freedom (such as a Cartesian coordinate or a bond length). This move is automatically accepted if the change in energy of the system, ΔU, is negative. If ΔU is positive then the Boltzmann factor, $\exp(-\Delta U/k_B T)$, is compared to a random number between 0 and 1. If the Boltzmann factor is higher than the random number then the move is accepted, otherwise the system is returned to the state before the rejected move and another random move is attempted. In this way, the ensemble of states generated by the Metropolis MC technique is weighted such that it comes from a Boltzmann distribution. A property of the system can be estimated by taking the simple mean of the property calculated over all the accepted states.

2.2: Conformational Transition Methods

Conformational transitions in biological macromolecules (such as proteins) are both an important and challenging field of study. The biochemical function of a protein can be dependent on some form of conformational transition. Well-known examples include the probable transition of the prion protein from a benign to an infective form,[9, 10] the opening and closing of the binding pocket in HIV protease,[11] allosteric effects such as in the *trp* repressor of *E. coli*,[12] and indeed, the protein folding process itself.[13]

If we are interested in simulating the real-world trajectory of a conformational transition in a protein, then molecular dynamics initially appears to be the obvious choice. However, traditional MD simulations cannot simulate many important conformational transitions in proteins because they happen over too long a time-scale. The motion of the flaps of HIV protease is thought to take place on the millisecond time-scale[14] and protein folding takes place on the time-scale of milliseconds to seconds or even longer. The longest protein simulation performed so far[15] is of just 1 μs and most are of just a few nanoseconds. In this limited amount of simulation time, adequate sampling of

conformational transitions cannot be achieved.

Therefore, the rate of conformational transitions must be increased if they are to be studied by these methods. This could be achieved by raising the simulation temperature, but high-temperature MD[16] does not model the protein at body temperature, and it is not clear whether conformational transitions at high temperature bear any relation to those occurring in physiological conditions.

How can we improve the conformational sampling in an MD simulation? What follows is a brief overview of some of the various techniques available for this purpose.

2.2.1: Increasing the Amount of Time Simulated

Perhaps the most obvious way of increasing the sampling in MD is to somehow increase the execution speed of the MD program so that more time can be simulated. This can be achieved in two ways: by changing to a faster computer or by improving the efficiency of the MD calculation. Since the late 1970s, the speed of computers has doubled roughly every 18 months (giving an increase in power of approximately 25 000 in the last 22 years). It is a combination of this extraordinary increase in computational capacity and the popularisation of parallel computing that has brought the field to the point where a nanosecond time-scale protein simulation is the norm rather than the exception.

The recent development of the Beowulf class of computer, where a parallel machine is constructed from commodity PCs and a high speed network, has led to parallel computers becoming more commonly available. A parallel computer splits a single calculation across several CPUs, increasing the speed of the calculation many times. Many popular MD packages (e.g. CHARMM, AMBER, DL_POLY and DLPROTEIN, refs. 5, 17–19) can be run on these parallel machines and new packages and methods[20] have been developed with modern distributed-memory parallel computers in mind. It seems certain that the proportion of MD calculations performed on parallel machines will increase still further.

Although the power of computers has increased so dramatically, it will still be 15 years at the current rate of increase, before microsecond time-scale simulations are as common as nanosecond ones today. By using efficient MD algorithms, it is possible to simulate for a longer time without waiting for a faster computer. The part of an MD calculation that takes the most time is the force calculation and, in particular,

the calculation of long-range electrostatic interactions. For a system of N particles, the number of these interactions scales with order N^2. This makes a direct calculation of the electrostatic forces impractical for a large system. The traditional approach to this problem has been to truncate the electrostatic calculation, giving an algorithm of order N. The electrostatic force scales with distance, r, as $1/r$, which means that any cut-off that is small enough to give a fast simulation, will produce a large error in the electrostatic calculation and result in an inaccurate simulation,[21] particularly if the system has many large charges such as a nucleic acid. It is now becoming common to use a method derived from the Ewald sum[22] such as particle mesh Ewald[23] (PME) or smooth particle mesh Ewald[24] (SPME). The Ewald summation computes the exact electrostatic interaction and can be written such that it is an order $N^{3/2}$ algorithm. Although this is a lower order than the direct electrostatic calculation, it is still prohibitively expensive for a large system. PME and SPME are approximations to the Ewald method that are both order $N \log(N)$ and are the most efficient known algorithms of this type. The PME method has been found to be as quick as a direct calculation with a 9 Å cut-off for a medium level of accuracy.[24]

The time-step in an MD simulation is limited by the fastest motion present. In a protein simulation, the fastest motion is the vibration of the hydrogen atoms.[25] The SHAKE algorithm[8] may be used to tightly constrain any atom-atom distance in the system. It operates at each time-step by considering each constraint in turn and adjusting the atomic positions to satisfy the constraints. One adjustment may mean an earlier constraint is no longer satisfied, and so the algorithm iterates until all constraints are satisfied within a given tolerance. By constraining all bonds in a protein to their equilibrium values, the fastest motion is still that of the hydrogen atom vibrations, but reduced from a bond vibration frequency to the frequency of valence angle motion. By applying the SHAKE algorithm in this way, the time-step may be raised from the normal 1 fs to 2 fs—doubling the speed of the calculation.

In protein molecular dynamics, motion on a wide range of time-scales needs to be sampled correctly. The reference system propagator algorithm[26] (RESPA) increases efficiency by using two different time-steps. The fastest motions (bond vibrations and valence angle bending) are integrated using a small time-step and the slowest motions

(non-bonded forces) are integrated using a large time-step. By using a large time-step for the non-bonded forces, the algorithm performs the expensive long-range force calculation much less frequently than in normal MD. More recently, the RESPA method has been reformulated[27] to provide a time reversible integrator and include Nosé thermostats. This method, r-RESPA, has been included in the ROAR MD program.[28] The RESPA method has been found to produce erroneous forces if the long time-step happens to coincide with a natural frequency of the system.[29] More complicated variations of the idea, using up to four different time-steps have been developed and tested.[30] The computational speed-up over normal MD by using one of the multiple time-step (MTS) methods is approximately 5-fold. A review of long time-step methods can be found in a paper by Schlick *et al.*[31]

The methods presented above may be applied to generic molecular dynamics simulations. The following two sections consider algorithms specifically designed to improve conformational sampling in molecular dynamics.

2.2.2: Methods for Known End-Points

In some cases, two conformations of a protein are known. The following techniques use various methods to find a reasonable path from one conformation to the other.

The 1987 technique of Elber and Karplus[32] (EK) finds the reaction path between two conformations that minimises the average potential energy along the path. Czerminski and Elber[33] improved the path searching speed of the EK technique and constrained rigid body rotation and translation to give the self penalty walk algorithm (SPW). The SPW algorithm has been further modified by Ech-Cherif El-Kettani and Durup[34], who calculate the reaction path in a set of internal coordinates. The improved algorithm has been applied to determine a conformational path between the closed and open forms of pig heart citrate synthase.

The Path Energy Minimisation (PEM) technique of Smart[35] is another development of SPW. A path is constructed between two fixed configurations using several adjustable configurations. A function that approximates the maximum potential energy along the path is then minimised by the adjustment of the intermediate points. The method has been applied to gramicidin A[35] and α-D-xylulofuranose.[35, 36]

The Conjugate Peak Refinement[37] (CPR) method of Fischer and Karplus calculates a reaction path between two known conformers that follows the adiabatic path and proceeds via true saddle points. The initial reaction path is taken to be the direct path between the two conformers. This is then refined using the gradient of the potential energy until the final path is reached. CPR may also be used to refine an initial guess at the reaction path.

All the above methods[32, 34, 36, 37] produce a single reaction path from the start-point to the end-point. While this may be useful for some purposes, it does not provide a complete picture of all the transitions possible in the course of true dynamics.

Targeted molecular dynamics[38] (TMD) is a method first published in 1993 (ref. 39), designed to induce conformational change to a known target structure at ordinary temperature by the application of a minimal geometric constraint. A transition in insulin was studied in the original paper and since then the method has been applied to a variety of systems including further insulin studies,[38, 40] decalanine,[38] the Ha-*ras*-p21 protein,[41] the activation of α-chymotrypsin,[42] and the folding and unfolding of chymotrypsin inhibitor 2 (ref. 43, 44).

The TMD method forces a trajectory from a start conformation, I, to a final (known) conformation, F. If the vector $\mathbf{x} = (x_1, \ldots, x_{3N})^{\mathrm{T}}$ contains the $3N$ Cartesian coordinates of the atoms and $\mathbf{x}_F$ is the vector for the final conformation, then one may define for each configuration, $\mathbf{x}$, the distance, ρ, to the target configuration:

$$\rho = |\mathbf{x} - \mathbf{x}_F| = \left[\sum_i (x_i - x_{Fi})^2\right]^{\frac{1}{2}} \tag{2.6}$$

An additional constraint is introduced which causes ρ to monotonically decrease as the simulation progresses. The rate of decrease of ρ is in inverse proportion to the simulation length, such that at the exact end of the simulation $\rho = 0$ and $\mathbf{x} = \mathbf{x}_F$.

The constraint may be termed a "minimal geometric constraint" because, in effect, only one of the $3N$ degrees of freedom is constrained. All other degrees of freedom are able to explore the configurational space. Two simulations are commonly performed: one moving from the start conformation to the end conformation and one moving from the "end" conformation to the "start". These two simulations, though tightly constrained at their ends, can diverge markedly in the mid-section.

Path exploration with distance constraints[45] (PEDC) is a technique that is very similar to TMD. In PEDC, the distance, ρ, from a reference structure, R, is measured in a similar fashion to TMD but with the addition of atomic masses, m_i:

$$\rho = \left[\frac{\sum_i m_i (x_i - x_{Ri})^2}{M} \right]^{\frac{1}{2}} \tag{2.7}$$

where M is the total mass of the atoms. By an addition to the potential energy equation, this mass-weighted root mean square distance may be used to attract a simulation towards a final conformation, as in TMD, to drive a simulation away from the starting conformation, or both. Two further constraints are also added to prevent the structure from rotating or translating with respect to the starting structure. PEDC may be used just to drive a simulation away from an initial structure so it could also be included in the following section of this review. TMD could also be reformulated to be used in this way.

Santos *et al.*[46] used a combination of PEDC and CPR to find a transition path from the unbound to the calcium-bound state of domain III of annexin V (ref. 46). In this paper, the initial guess is formed using the PEDC method and then refined using CPR.

Adiabatic bias molecular dynamics[47] is another technique similar in concept to TMD and PEDC. A time-dependent bias potential, V_{ab}, is added to the system potential:

$$V_{\mathrm{ab}}(t_n) = \alpha [R(t_n) - G_0(t_n)]^2 \tag{2.8}$$

where $R(t_n)$ is some reaction coordinate at time-step t_n and the bias parameter, G_0 is defined as:

$$G_0(t_n) = \begin{cases} G_0(t_{n-1}), & \text{if } R(t_n) < G_0(t_{n-1}); \\ R(t_n), & \text{if } R(t_n) \geq G_0(t_{n-1}). \end{cases} \tag{2.9}$$

In other words, the bias parameter gently but constantly, pushes the simulation towards a larger value of the reaction coordinate. If the perturbation due to V_{ab} is kept small by a good choice of α, then the system will move on an almost constant energy surface and the transformation will be adiabatic.

Conformational Transitions by Molecular Dynamics with minimum biasing[48] (CONTRA MD) requires a measuring function, P, that measures progress along the pathway

of the transition being studied. This could be the function used in TMD or PEDC, or something entirely different involving the radius of gyration for instance. The function must be single valued for all configurations and, if possible, differentiable with respect to the atomic coordinates. To perform CONTRA MD, a series of short (e.g. 0.1 ps) MD simulations are run. At the end of each simulation, the algorithm decides whether to accept or reject that portion of the simulation. The value of P is evaluated and compared with its value at the beginning of the short simulation. If progress has been made towards the target value of the function, then the simulation is accepted and another short simulation is performed, starting at the end-point of the accepted simulation. If no progress has been made, then the short simulation is rejected. The system is then reset back to the configuration at the beginning of the rejected simulation and the velocities are randomly reassigned from the Maxwell-Boltzmann distribution. Another short simulation is then initiated from this new state. The procedure is repeated until the target value of P is reached. The resulting trajectory will be continuous in its coordinates but will have sharp changes of direction.

A variation of the basic CONTRA MD method described in the same paper,[48] is to choose the random velocities such that the acceptance rate of the simulations is increased. When the algorithm comes to assign random velocities, a series of random sets of velocities are generated and each one is evaluated (using the coordinate derivative of P) to see which one will produce the most initially favourable change in P. This velocity set is chosen and the simulation continued as before.

TMD, PEDC, adiabatic bias MD and CONTRA MD all use a function to measure the progress of the simulation from one state to another. All require that progress is made at every measurement point along the trajectory. Although the functions used in these methods constrain the system by a minimum amount, it is still a constraint and as such removes all but a few possible paths from consideration. There can be no monotonically increasing measurement of progress that is applicable to all conformational change events, so these methods must be used with care.

The stochastic path approach of Elber *et al.*[49] computes approximate optimised trajectories from the start point to the end point by using very large time-steps and approximating the high frequency motion by Gaussian white noise. The method has

been applied to several systems including alanine dipeptide[49] and C peptide.[50]

Finally, in steered molecular dynamics (SMD), though the end point itself is not known, the direction of conformational change is defined by an external force which is applied to the system as the dynamics evolves. Commonly, this force takes the form of a harmonic spring attached to part of a molecule for the purpose of simulating the unbinding process (such as retinal from bacteriorhodopsin,[51] or lipid from a bilayer[52]), or for investigating how a molecule stretches (such as the unfolding of titin immunoglobulin domains[53]). The computer experiments are similar in nature and result to atomic force microscopy (AFM) but, due to the limitation of computer power, are forced to proceed 10^6 times faster than the AFM experiment.

2.2.3: Methods Requiring No Knowledge of the Final State

In most cases, only a single structure of a protein has been experimentally determined. To investigate conformational change in these situations requires methods that do not pre-determine the end-point of the simulation—methods that just enhance conformational change in a general sense. Since the amplitude of a mode of motion is inversely proportional to its frequency,[54] the most significant motion is produced by the low frequency modes. Of the methods presented here, SEMD and SGMD both aim to enhance the low-frequency motion of the system. Conformational flooding, PCR-MD and LD all use essential dynamics,[55] or a variant, as a basis for driving conformational change. Essential dynamics analyses a normal MD trajectory and extracts the so-called "essential modes". The first few of these modes often succinctly describe a large proportion of the motion in the system.

The method of Selectively Enhanced Molecular Dynamics (SEMD) has only appeared in the literature as a small section in a paper that was primarily focused on the analysis of MD trajectories.[56] However, this method is an important precursor to the techniques described in this work and so is discussed both here and later in the thesis (see section 4.1). The SEMD method alters the dynamics of the system by continuously replacing the velocity set with the average of the velocity sets from the previous n time-steps. The averaging has the effect of reducing high-frequency motion and enhancing low-frequency motion. The velocity of an atom vibrating quickly will have changed

sign at least once during the n steps, and so by the application of the averaging window, velocities of high-frequency motion will be reduced or even cancelled completely. The velocity of an atom that vibrates very slowly may not change sign at all during the n time-step window, and so the substitution of the average velocity enhances the low-frequency motion. The choice of n determines which frequencies are enhanced and which are reduced. There are several problems with this technique that are discussed in section 4.1.

The self-guided molecular dynamics (SGMD) method was first described in 1998 (ref. 57). It is a method of enhancing the rate of conformational change without any predetermined target conformation. The method has been applied to alanine dipeptide,[57] liquid argon,[58] a 16-residue synthetic peptide,[57, 59] and the linear pentapeptide YPGDV (ref. 60). Recently, the algorithm has been re-formulated so that as well as working in the micro-canonical and canonical ensembles, it can be used in the isothermal-isobaric ensemble.[61]

In SGMD, a guiding force is derived from the average non-bonded forces present in the simulation trajectory directly preceding the current time-step. This guiding force is re-calculated at each time-step and incorporated into the equations of motion to enhance the systematic motion of the system. An SGMD simulation requires two parameters: one is the length of time over which to perform the average, which, for proteins in explicit water has commonly been chosen to be between 0.2 and 2 ps. The second parameter is a scaling factor for the guiding forces and defines how much guiding effect will be introduced into the simulation. It is commonly set to between 0.2 and 0.5 (ref. 59).

Conformational Flooding[62] may be thought of as a generalisation and modification of the umbrella sampling idea more commonly applied to Monte Carlo simulations. An initial MD simulation is performed and analysed using essential dynamics.[55] This analysis provides a set of modes that describe the essential dynamics of the system during the trajectory. The most harmonic, high-frequency modes are discarded and a "flooding potential" is derived to reduce the well-depths in the potential surface of the remaining coordinates. MD simulations are then performed using the potential modified by the flooding potential. By decreasing the well-depths, the rate of conformational change is increased. The degree to which the well-depths need to be "flooded" is not

known, so several simulations with different scale flooding potentials must be carried out. Conformational flooding has recently been applied to myoglobin,[63] which is the first protein system to be studied by this technique.

Principal component restraint molecular dynamics[64] (PCR-MD) applies a restraint to an ensemble of otherwise independent MD simulations. As with conformational flooding, an initial MD simulation is performed and analysed using essential dynamics. A number of MD simulations are then run in parallel and the standard deviation (and optionally the average) of each essential-mode coordinate is calculated and restrained to a target value using an additional energy term. Restraining the standard deviation of the ensemble to a large value has the effect of promoting diversity among the parallel simulations and thus increases the conformational sampling. The method has been applied to the dynamics of the PH domain of β-spectrin.[64] In this work a 9 ns initial simulation was performed followed by ten parallel 10 ps PCR-MD simulations. The conformational space coverage of the PCR-MD simulations was found to be comparable to the initial 9 ns simulation and new areas of conformational space were explored.

The method of essential dynamics sampling (EDS) by Amadei *et al.*[65] uses the coordinates of an essential dynamics analysis to drive a MD trajectory. In the original paper[65], an initial 600 ps MD simulation was performed and the essential coordinates were extracted. The first, third and fourth essential coordinates were chosen to drive the position of the EDS simulation by measuring the distance from the start point along the coordinates and constraining this radius. A cycle of three parts was used: 5000 constrained steps where the radius expanded by 0.0004 nm each step, 5000 constrained steps with the radius fixed and then 20 000 steps of free MD to sample the local configurational space. The final configuration is then taken to be the centre of the next expansion sphere and the cycle is repeated. This protocol was applied to histidine-containing phosphocarrier protein (HPr) and was found to increase the sampling rate.

A modification of the method was presented shortly afterwards.[66] The protocol was improved so that the system was not forced to expand away from the start point along the radius of the essential coordinates at every step. Instead, a normal MD step is taken and accepted if the radius has expanded. However, if the new position is closer to the start point (the radius has reduced) it is moved back by a force constraining the

essential coordinates such that the radius is kept constant. In this way, the simulation can explore configurations at a constant distance from the start point until it finds a pathway that permits the distance from the start point to increase. If no expansion occurs for a certain time (500 steps in their example) then a new expansion cycle is initiated with the final position of the old cycle taken as the new centre. The modified method was applied to the peptide hormone guanylin.

Conformational flooding, PCD-MD and EDS all require a long MD simulation to be performed for the purpose of extracting the essential modes. Leap-dynamics[67] (LD) overcomes this computational expense by calculating these prominent structural variations using the CONCOORD essential dynamics[68] (CED) method. CED generates an ensemble of realistic structures from an initial configuration by first generating a complete set of atom-atom distance constraints based on the initial configuration. The atom-atom distances are allowed to vary according to their interaction type. For instance, the variation allowed in a covalently bonded atom pair is 0.002 nm, 0.005 nm in a 1–3 pair, 0.02 nm for a 1–4 pair in a "loose" ϕ or ψ peptide dihedral or 0.5 nm for non-interacting atoms. Given these distance constraints, an ensemble is generated by choosing random structures and adjusting them iteratively until they fulfill the constraints. Once several hundred structures have been generated, the essential dynamics technique is applied to the ensemble and the usual essential modes are extracted. It is these modes which are used in LD to increase conformational change using a combination of driving the simulation along the essential modes, energy minimisation and molecular dynamics. LD has been applied to alanine dipeptide and bovine pancreatic trypsin inhibitor (BPTI).

Locally Enhanced Sampling[69] is well suited to enhancing the sampling of systems where a ligand is in complex with a larger molecule. The larger molecule (e.g. a protein) is simulated once, while multiple simulations of the ligand are simultaneously performed. The protein experiences the average force of the ligands and each ligand experiences the full force of the protein. The rate of sampling is increased because the expensive part of the calculation–the larger protein–is only simulated once for multiple copies of the ligand. This method is not suited to systems where the protein conformation is significantly affected by the position of the ligand[70] as the protein's response to each

ligand is a small fraction of what it should be. LES has also been used to enhance the sampling of part of a small RNA hairpin loop. In this application,[71] LES was combined with the particle mesh Ewald method and five copies of the four-base part of the hairpin were simulated to successfully enhance the conformational change with respect to a normal MD simulation.

2.3: Conclusion

In this chapter, both generic methods for increasing the length of MD simulations and methods specifically designed to improve conformational sampling have been described. Many of the generic methods for increasing the simulation length (such as particle mesh Ewald and RESPA) are implemented in recent versions of the common MD packages and thus are being increasingly used and accepted.

The methods that require knowledge of both conformational end-points are of limited use because it is unusual to have more than one distinct structure available. The EK method and its derivatives produce only a single reaction path which gives no information on the true dynamics of the system. The methods using an extra constraint (TMD, PEDC, CONTRA MD and SMD) all unavoidably limit the reaction path to one in which the constraint is (perhaps unrealistically) monotonically varied and so must be used with care.

Of the methods in which the end point is not required, SEMD and SGMD both aim to enhance conformational sampling by enhancing the low frequency motion in the system—the approach used in this thesis. Conformational flooding, PCR-MD and EDS all use essential dynamics (ED) as a basis for driving the conformational change. There is a problem with this approach in that firstly a long MD simulation must be performed to calculate the essential modes. As well as being computationally expensive, there are concerns[72] that the ED method is not suitable for use with protein systems: the fundamental problem is that the modes produced by the ED analysis are only characteristic of the simulation analysed, and can change from one length of simulation to another. As has already been discussed, the initial MD simulation cannot be long enough to explore the full character of the real-world motion, and so the ED modes are incomplete and variable. The benefit of using these modes in driving conformational

change is therefore limited.

3: Digital Signal Processing

In this thesis, the application of digital signal processing (DSP) techniques to the analysis and control of molecular dynamics simulations will be explored. Digital signal processing is concerned with manipulating and analysing information that is measured as discrete sequences of numbers. The continuing increase in computer power has led to a corresponding increase in the use of DSP methods. Specialist DSP micro-chips are now found in telephones, fax machines, audio equipment and computer graphics cards as well as more specialised applications such as sonar and radar equipment.

Digital signal processing techniques work with discrete sequences of numbers, not directly with continuous data. A molecular dynamics trajectory provides many such discrete sequences of data such as the variation of the coordinate and velocity of each atom with time.

In this chapter, the aspects of digital signal processing important to this work are presented.

3.1: Sampling

If the data source is continuous then it must first be sampled before being fed into the DSP algorithm. It is important that the data is sampled fast enough to avoid

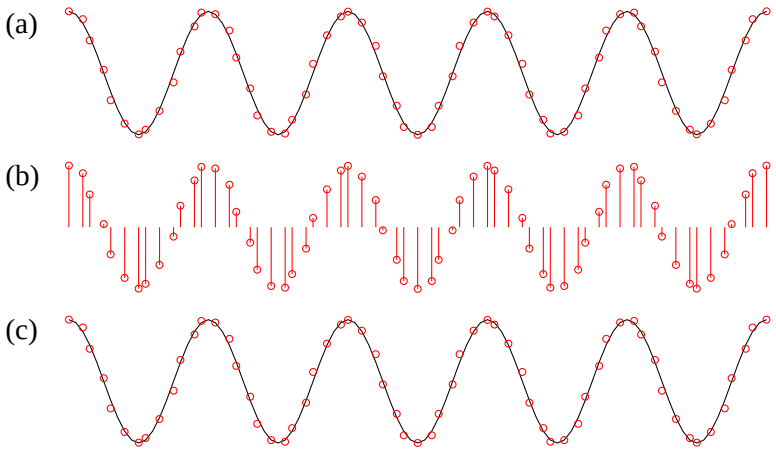

Figure 3.1. Digitally sampling a wave: (a) The 5 Hz wave is sampled at 62 Hz (b) The sampled points (c) The wave is reconstructed from the sampled points.

a phenomenon known as "frequency folding" or "aliasing". In a molecular dynamics simulation we use the approximation that the data (coordinates, velocities, etc) at each time-step is a sample from a continuous motion. Therefore our choice of time-step determines the maximum sampling rate of the data. This maximum possible sampling rate will invariably be fast enough for DSP purposes but it is important to understand the effects of under-sampling the data. When a trajectory is dumped to disc it will often be written or "sampled" periodically rather than every step. In this case the sampling issues become important.

Figure 3.1 shows a wave that is being sampled sufficiently to be reconstructed from the sampled data points correctly. Figure 3.2 shows the same wave being sampled at a much lower sampling rate. Owing to the low sampling rate, when the wave is reconstructed in 3.2(c), the original wave has not been reproduced. A lower frequency wave fits the sampled data points just as well, and the high frequency is said to have been "aliased" or "folded".

As an example, let us take a cosine wave oscillating with frequency ν Hz. The angular velocity is $\omega = 2\pi\nu$ rad s^{-1}. If our sampling frequency is f_s Hz then our sampling period, $t = 1/f_s$ s. We will therefore be taking a sample, p_k, once every $t\omega$ rad (where k is an integer). This quantity is known as the normalised frequency.

$$p_k = \cos kt\omega \tag{3.1}$$

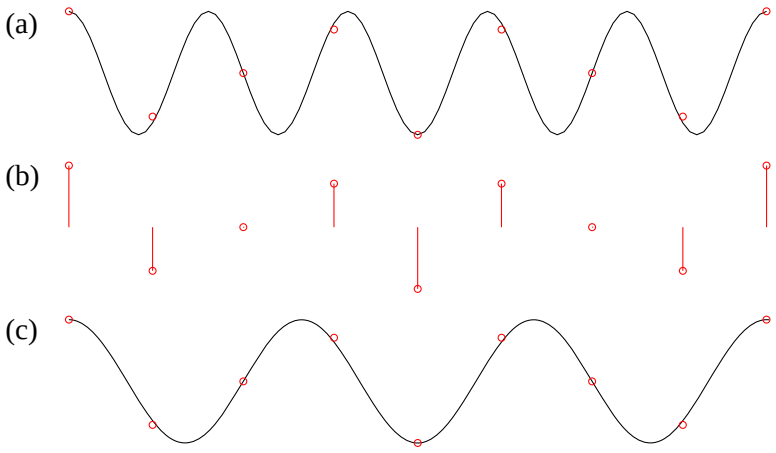

Figure 3.2. Digitally sampling a wave: (a) The 5 Hz wave is sampled at 8 Hz (b) The sampled points (c) A lower frequency wave (3 Hz) is constructed from the sampled points.

We can express the normalised frequency, $t\omega$, as the sum of two terms:

$$t\omega = \omega' + 2\pi n, \qquad \text{where} \qquad |\omega'| < \pi \qquad \text{and} \qquad n \in \text{integer set} \tag{3.2}$$

Equation 3.1 can then be rewritten to incorporate equation 3.2:

$$p_k = \cos(k(\omega' + 2\pi n)) = \cos(k\omega' + 2\pi nk) \tag{3.3}$$

As k and n are integers and cosine is periodic in 2π, the $2\pi nk$ term can be discarded to leave equation 3.4 as the final sampling equation:

$$p_k = \cos k\omega' \tag{3.4}$$

This means that the samples of a cosine wave depend only on the portion of the normalised frequency that is less than π.

If we sample a 5 Hz wave at 8 Hz as in figure 3.2 then the normalised frequency is $5 \times 2\pi/8$ rad/sample. The magnitude of this is greater than π. If we subtract 2π then we have $-3 \times 2\pi/8$ rad/sample as an equivalent normalised frequency. This means that the 5 Hz wave sampled at 8 Hz will give the same samples as a -3 Hz wave sampled at 8 Hz.

How can a frequency be negative? It is purely a convenient mathematical concept. The magnitude of a -3 Hz wave is the same as a 3 Hz wave and the phases are equal but with opposite sign. In other words, they are the same oscillation but one is spinning in the opposite direction to the other. We could have taken the range for the normalised

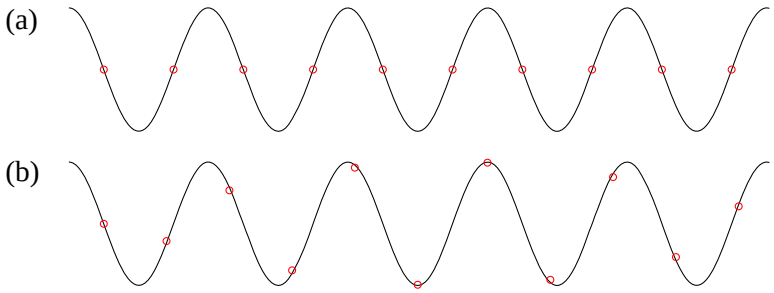

Figure 3.3. Digitally sampling a wave: (a) The 5 Hz wave is being sampled at the Nyquist critical frequency of 10 Hz (b) The sampling rate is 11 Hz.

frequency to be any continuous 2π region but using $-\pi$ to π is the most convenient and elegant choice.

In order to sample a frequency ν with no ambiguity, we must ensure that n in equation 3.2 is zero, that is $|t\omega| < \pi$:

$$|t\omega| < \pi \quad \Rightarrow \quad t < \frac{\pi}{|\omega|} \quad \Rightarrow \quad \frac{1}{f_s} < \frac{\pi}{|\omega|} \quad \Rightarrow \quad f_s > \frac{|\omega|}{\pi} \quad \Rightarrow \quad f_s > 2\nu \qquad (3.5)$$

So our sampling frequency must be greater than twice the frequency of our waveform. Or, put another way, the oscillation should not travel more than half way round between samples. This limit is known as the Nyquist critical frequency, ω_N. Figure 3.3(a) shows a wave being sampled at exactly ω_{N}. The worst case is shown, where the relative phase of the wave and the sampling means that all the samples fall at zero amplitude. In this case, the original wave cannot be reconstructed from the samples. Figure 3.3(b) shows the same wave being sampled at a frequency just greater than ω_{N}—in this case the data is sampled sufficiently and the original wave can be reconstructed.

At each step in a molecular dynamics simulation, the positions, velocities and forces of the atoms are sampled. If the time step is t s then, by similar arguments, no motion faster than $1/2t$ Hz can be perceived. For a time step of 2 fs, the fastest motion we can observe is 8339 cm^{-1}. In protein dynamics, the fastest motion is the hydrogen atom motion with a period of approximately 10 fs which is 3336 cm^{-1} (ref. 25) so we can of course observe this fastest motion with no problem. If, however, a molecular dynamics trajectory is stored for later analysis then the time between saved steps must not be too long. For a fully flexible protein system the dumping interval must be no longer than 5 fs, and for one with the bonds constrained the interval should be no more than

10 fs, as the fastest motion in this case—the angle vibration of a hydrogen atom—has a period of approximately 20 fs (ref. 25). If the sampling rate is not high enough, then any frequencies higher than half the sampling frequency will be aliased and will appear incorrectly as lower frequencies in any analysis.

3.2: Spectral Density

An atom in an MD simulation vibrates with many frequencies. A plot of a component of the atom's coordinate or velocity will in general show a complicated waveform due to the superposition of these frequencies. It is useful to analyse a molecular dynamics trajectory to find out what frequencies of motion are present. One method of doing this is to calculate the spectral density.

The spectral density is calculated using a Fourier transform of the atomic velocity or coordinate components.[73–75] This has the advantage that none of the harmonic approximations associated with normal mode analysis are used and that the spectra obtained are based on entire MD trajectories. The spectral density, $g(\nu)$, can be thought of as the simulation equivalent of an IR vibrational spectrum. It shows the relative amount of energy (or density of states) in each frequency, ν Hz. The integral of $g(\nu)$ is theoretically the total number of degrees of freedom in the system.

For a single particle of mass m, oscillating in just the x dimension, the formula for the $g(\nu)$ is as follows:

$$g(\nu) = \frac{1}{k_{\mathrm{B}}T} m \left(2\pi\nu \int_{-\infty}^{\infty} x(t) \mathrm{e}^{-2\pi i \nu t} \, dt \right)^2 \tag{3.6}$$

Where k_{B} is the Boltzmann constant, T is the temperature in Kelvin, $x(t)$ is the displacement of the mass from its equilibrium position at time t and i is $\sqrt{-1}$. This equation is derived in appendix A.1. For a system of n particles oscillating in three dimensions, we can use a more general equation:

$$\begin{aligned} g(\nu) = \sum_n g_n(\nu) = \sum_n \frac{1}{kT} m_n & \left(2\pi\nu \int_{-\infty}^{\infty} x_n(t) \mathrm{e}^{-2\pi i \nu t} \, dt \right)^2 \\ & + \left(2\pi\nu \int_{-\infty}^{\infty} y_n(t) \mathrm{e}^{-2\pi i \nu t} \, dt \right)^2 + \left(2\pi\nu \int_{-\infty}^{\infty} z_n(t) \mathrm{e}^{-2\pi i \nu t} \, dt \right)^2 \end{aligned} \tag{3.7}$$

Or, more briefly:

$$g(\nu) = \sum_n g_n(\nu) = \sum_n \frac{1}{kT} m_n \left(2\pi\nu \int_{-\infty}^{\infty} x_n(t) \mathrm{e}^{-2\pi i \nu t} \, dt \right)^2 \tag{3.9}$$

where the sum across the dimensions is implicit. Expression 3.9 is in terms of the particles' masses and displacements over time. It is also useful to be able to calculate $g(\nu)$ in terms of the particles' velocities. The equation for this is derived in appendix A.2 and is presented here:

$$g_n(\nu) = \frac{1}{kT} m_n \left(\int_{-\infty}^{\infty} \dot{x}_n(t) \mathrm{e}^{-2\pi i \nu t} \, dt \right)^2 \tag{3.9}$$

A variation of the spectral density may be used to focus on the motion in an internal coordinate, such as a single bond or dihedral. In order to do this, the internal coordinate must first be calculated as a function of time from the stored Cartesian coordinate trajectory. A sophisticated analysis program has been written to extract any internal coordinate from a trajectory. Neither spectral density equation given so far is appropriate to this situation because both require the mass of the oscillator. For an oscillating diatomic, the reduced mass may be used, but for any larger system the reduced mass equivalent is not known. However, for the case of a single internal coordinate trajectory, the mass merely scales the data in the y-axis and so can be removed from the equation along with the other scaling factors. The spectral density equation for a single degree of freedom is therefore:

$$g(\nu) = \left(\nu \int_{-\infty}^{\infty} x(t) \mathrm{e}^{-2\pi i \nu t} \, dt \right)^2 \tag{3.10}$$

In general, the magnitude of the peaks in a spectrum calculated from this equation cannot be compared quantitatively to another spectrum because the mass term and the other scale factors have been removed. Relative peak heights within a single spectrum can, however, safely be compared.

When an MD simulation is run for the purpose of calculating a spectral density, the velocities or coordinates must be saved often enough to ensure good sampling (see section 3.1) and the simulation must also be run for long enough. The sampling period determines the maximum frequency. The Nyquist criterion given in equation 3.5 says

that to observe a frequency ν with no ambiguity, the frequency of our sampling must be at least twice ν. Or, to put it the other way round, if we sample with frequency f_{sample}, then we can observe frequencies up to (but not including) $f_{\text{sample}}/2$. If our sampling period is $t\,\text{s}$, then our sampling frequency is $1/t\,\text{Hz}$. We can observe frequencies up to $1/2t\,\text{Hz}$ or, in wavenumbers:

$$f_{\max} = \frac{1}{200ct} \tag{3.11}$$

where c is the speed of light in a vacuum in $\text{m}\,\text{s}^{-1}$.

The total simulation time, T, determines the frequency resolution. No more data points can be calculated in the frequency domain than exist in the time domain. If the number of data points in the time domain is T/t, then T/t data points must be evenly spaced out in the frequency domain (positive and negative frequency) giving the equation:

$$\begin{aligned} \text{Frequency resolution} &= \frac{2f_{\max}}{T/t} \\ &= \frac{1}{100cT} \end{aligned} \tag{3.12}$$

To get a spectral density with $1\,\text{cm}^{-1}$ resolution, the simulation length must be approximately 33 ps. Initially it is advisable to simulate with a small sampling period to properly observe very high frequencies. Once the maximum frequency motion of the system is known, the inverse of equation 3.11 can be used to determine the largest sampling rate that avoids aliasing.

3.2.1: Discrete Fourier Transform

As we have seen, to calculate the spectral density we must employ the discrete Fourier transform. When interpreting the result of a discrete Fourier transform (DFT), we must bear in mind that the DFT is not perfect and that some of the signals in the spectrum may be false. The first type of false signal is caused by aliasing. This is where a high frequency is represented incorrectly as a lower frequency. Aliasing occurs if the sampling rate is not sufficiently high for the data, and has already been discussed in section 3.1. The second effect is "spectral leakage". This is where a single frequency in the input waveform is represented not as a sharp peak in the output, but as a broad peak, possibly

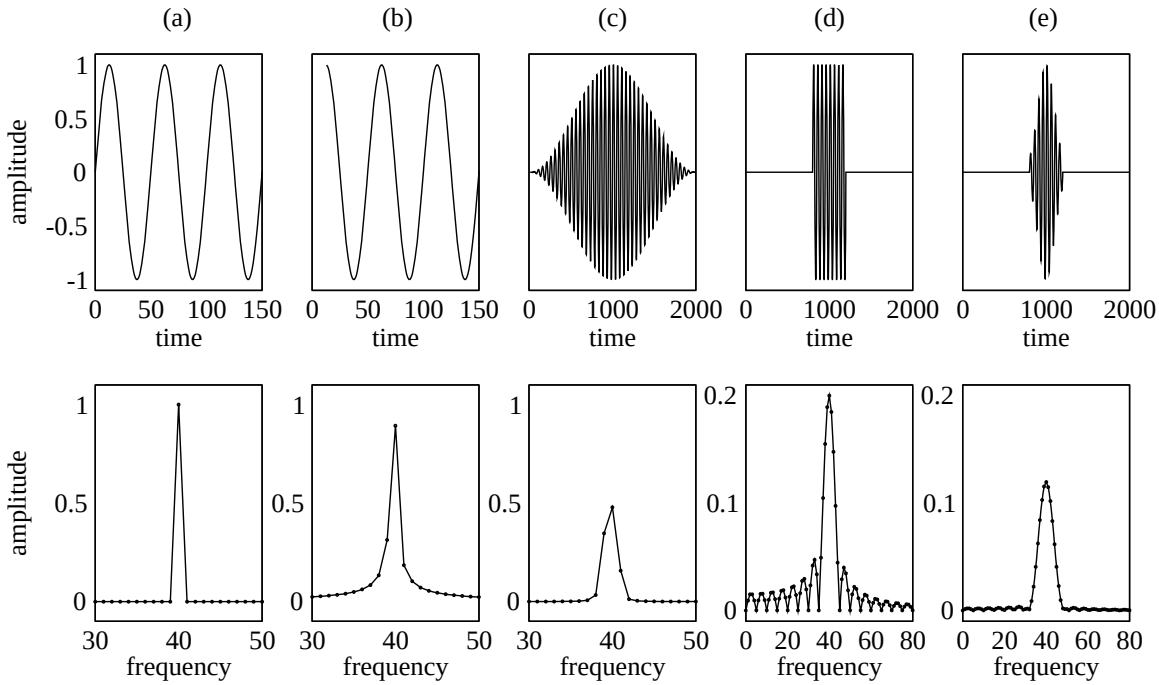

Figure 3.4. Five waveforms and their discrete Fourier transforms. The units of time and frequency are arbitrary but internally consistent. In part (a) and (b) only the first portion of the waveform is shown; the time axis for the full waveform extends to 2000. Note the change of scale in the frequency plots for parts (d) and (e).

with side-lobes. Spectral leakage occurs in two cases: when the total time sampled is not an integer multiple of the frequency's period, and if the frequency is localised in time.

Figure 3.4 shows five waveforms and their discrete Fourier transforms. Parts (a) and (b) show only the first few samples of the wave that was sampled up to "time" 2000. The Fourier transform of (a) is exact—it has an amplitude of zero at all points apart from the precise frequency of the wave where the amplitude is one. The DFT is precise because their are an integer number of complete periods of the wave during the time of the sampling.

In 3.4(b) the wave used in (a) was truncated such that there is no longer an integer number of periods; this is the usual case in a real life situation. The DFT of (b) has been broadened but the centre of the peak is still in the correct place (albeit with reduced amplitude). The frequency has been broadened because the DFT is formulated for use on periodic functions. The sample of the waveform we pass into the DFT equation is mathematically replicated in time out to infinity in both positive and negative directions.

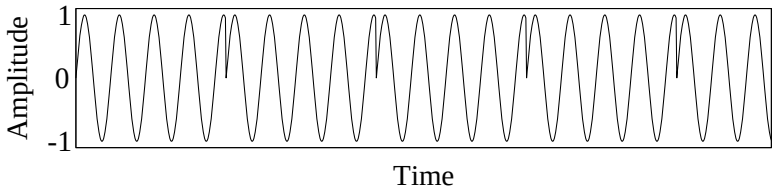

Figure 3.5. This shows an example of the type of signal that is Fourier transformed if there is not a whole number of periods in the sample being transformed.

In (b) there is not a complete number of periods in the sample, so by butting successive samples together a discontinuity is created, as shown in figure 3.5. The DFT can recreate this discontinuity by giving many more frequencies non-zero amplitudes, but the amplitude of the actual frequency must be reduced accordingly. If we take the square of the DFT then we have a plot of relative energy versus frequency (since the energy in a wave is proportional to the square of its amplitude). The integral of this energy function is unity in both parts (a) and (b), demonstrating that all the energy is represented but that the frequency has been blurred.

In part (c) the product of the same wave as (b) and the von Hann window function has been taken. The shape of the von Hann window is clear from the figure. It has the effect of gradually bringing the amplitude to zero at both ends of the sample. This means that there will be no discontinuity for the DFT to recreate when the sample is replicated in time. The DFT of (c) is still not a precise signal of course because now the DFT is having to use additional non-zero amplitudes to recreate the attenuating effect of the window. The von Hann window reduces the amplitude of the DFT peak but removes the long-range spectral leakage.

Figures 3.4(d) and (e) show the effect of a time-localised event. Part (d) shows the same frequency wave as (a) but only present for the middle 400 samples. The DFT of (d) has a lot of spectral leakage and shows long-range side-lobe bumps. The waveform in (e) is the same as (d) but attenuated by a Gaussian function. This shows the effect of a signal slowly being introduced and then removed. The peak is still very broad but has smaller side-lobes. Unfortunately there is no way to process the signal to remove the effect of time-localised events in the DFT method. Alternative methods of analysis are discussed in chapter 6.

As well as being used in the spectral density calculation, the DFT by itself can

provide useful information about the motion in a molecule. The DFT of a single degree of freedom provides a spectrum showing the amplitude of the motion at each frequency. It does not make any physical sense to sum more than one of these spectra so, in general, they must only be used on one degree of freedom at a time. As with the spectral density of the internal coordinate (which shows the relative amount of energy in each frequency but requires knowledge of the force constant), comparisons between spectra of different degrees of freedom must be made carefully and qualitatively.

In this work, spectral densities and amplitude spectra are calculated for various MD trajectories. To achieve a good frequency resolution, a long period of data must be analysed (see equation 3.12), but in all but the simplest of systems the frequencies present will change in intensity during the analysis period, producing time-localised events. Great care must be taken in interpreting these spectra therefore, as they do not represent a snapshot from a single time but rather contain information gathered over a time period. In figure 3.6 two waveforms are shown with their spectral densities. Both waveforms are composed of the same two frequencies: the first switches from one frequency to the other halfway through the time period and the second is the sum of the two frequencies (at half the amplitude) throughout the entire time period. Their spectral densities are almost identical—the only difference is that the first spectral density has some ripples due to the time-localisation of the frequencies. If the waveforms were representing atomic displacements, then the atoms' motions would be dramatically different. However this is not observed in the spectral density.

3.3: Digital Filters

A digital filter takes a series of input data (often a time series) and performs a mathematical function on a portion of the data to produce an output. Digital filters may be used to affect different frequencies in the input data in different ways. For instance, a digital filter in an electrocardiogram (EKG) machine is used on the data from the heart-sensor to remove high frequency noise but leave low frequency information, in order to see the signal from the heart more clearly. A bank of filters are also used in hi-fi systems to produce displays showing the volume in a range of frequency bands. Each filter in the filter bank acts on the output of the amplifier and is designed to remove all

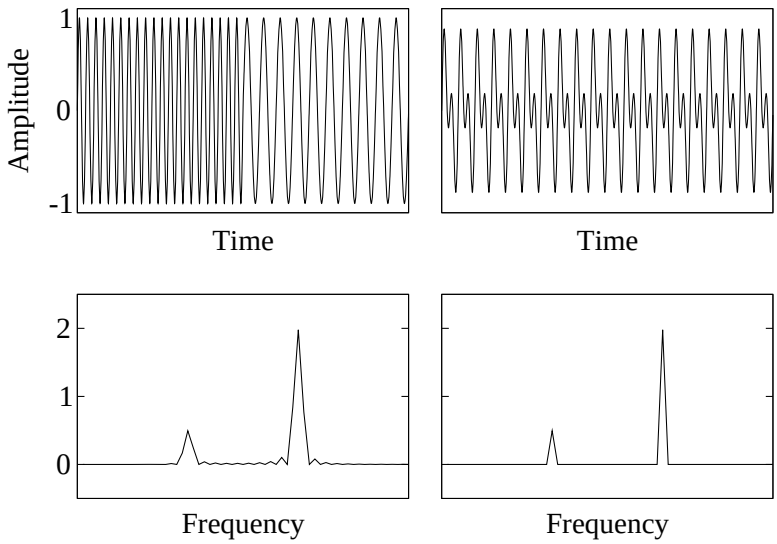

Figure 3.6. The top figures are two continuous waveforms. The bottom figures show the spectral densities of the waveforms.

frequency noise apart from in a narrow range. The output of each filter is measured to determine the intensity and hence the display light to activate.

Digital filters are used in the molecular dynamics applications presented in chapters 4 and 5 where they act on coordinate and velocity data series from the dynamics to alter frequency characteristics. Sub-sections 3.3.1–3.3.4 cover the function, application, design and properties of digital filters.

There are two main classes of filter, recursive filters and non-recursive filters. The input to a non-recursive filter is purely data from the series, but the input to a recursive filter is data from the series and previous filter outputs. The application of filters in this work is ideally suited to non-recursive filters and so recursive filters will not be discussed.

Consider the common case that the filter is being applied to a time series of data. The data could be being generated in real time, such as the sampled output of a microphone or the data could have been stored for later processing such as an MD trajectory. The kth value of the property we are filtering, p, will be written p_k. The non-recursive filter is a simple function of the time series. Each value of p is multiplied by a corresponding coefficient, c, and the sum of these products is the output, q_k:

$$q_k = \sum_{n=-m}^{m} c_n p_{k-n} \tag{3.13}$$

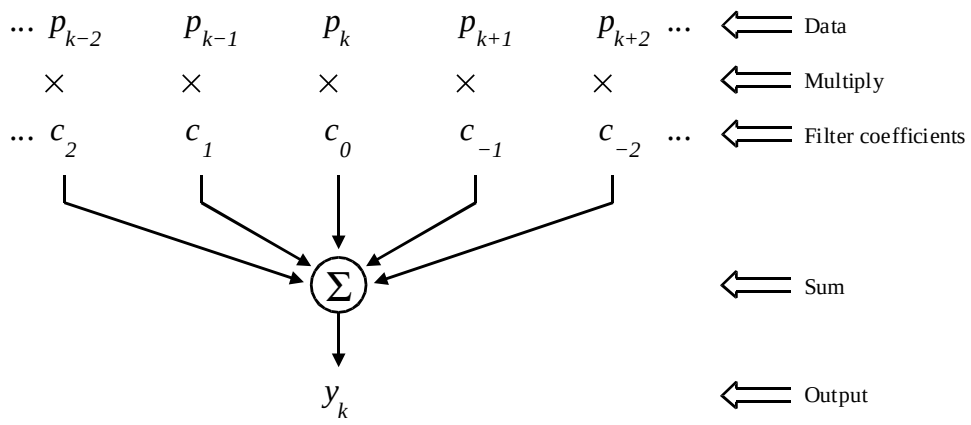

Figure 3.7. The application of a non-recursive digital filter.

Figure 3.7 shows this operation graphically. It is important to note the convention of corresponding the input p_{k+n} with the coefficient c_{-n} rather than c_n. The numbering of the coefficients from $-m$ to m and positioning of the output y_k opposite the middle input p_k are both arbitrary but convenient decisions. It means that when looking at the frequency response of the filter (section 3.3.1), the output will be compared with the mid-point of the section of data being filtered.

3.3.1: Frequency Response

The frequency response of a digital filter tells us how the filter will affect a given frequency input. We can find the frequency response by feeding an eigenfunction through the filter. The eigenfunction used is $\exp(i\omega k)$:

$$\begin{aligned} q_k &= \sum_{n=-m}^{m} c_n \mathrm{e}^{i\omega(k-n)} \\ &= \sum_{n=-m}^{m} c_n \mathrm{e}^{i\omega k} \mathrm{e}^{-i\omega n} \\ &= \mathrm{e}^{i\omega k} \sum_{n=-m}^{m} c_n \mathrm{e}^{-i\omega n} \end{aligned} \tag{3.14}$$

The eigenfunction has come through the filter intact but modified by the frequency response, $\mathcal{H}(\omega)$:

$$\mathcal{H}(\omega) = \sum_{n=-m}^{m} c_n \mathrm{e}^{-i\omega n} \tag{3.15}$$

Using Euler's formula $\exp(i\theta) = \cos\theta + i\sin\theta$ we can rewrite equation 3.15 in terms of sine and cosine:

$$\begin{aligned}\mathcal{H}(\omega) &= \sum_{n=-m}^{m} c_n \left(\cos(-\omega n) + i\sin(-\omega n)\right) \\ &= \sum_{n=-m}^{m} c_n \cos(\omega n) - i\sum_{n=-m}^{m} c_n \sin(\omega n)\end{aligned} \tag{3.16}$$

As $\mathcal{H}(\omega)$ is just a sum of $\cos(\omega n)$ and $\sin(\omega n)$ terms where n is an integer, $\mathcal{H}(\omega)$ is itself a periodic function with period 2π.

From equations 3.15 and 3.16 it is clear that $\mathcal{H}(\omega)$ can have a non-zero imaginary component. As it is a complex number, a useful way to consider the frequency response of a filter is to look at it in terms of magnitude and phase.

$$\begin{aligned}&\text{The magnitude} \quad |\mathcal{H}(\omega)| = \sqrt{\Re[\mathcal{H}(\omega)]^2 + \Im[\mathcal{H}(\omega)]^2} \\ &\text{The phase} \quad \phi(\mathcal{H}(\omega)) = \begin{cases} \tan^{-1}\left(\dfrac{\Im[\mathcal{H}(\omega)]}{\Re[\mathcal{H}(\omega)]}\right), & \text{if } \Re[\mathcal{H}(\omega)] \geq 0; \\ \tan^{-1}\left(\dfrac{\Im[\mathcal{H}(\omega)]}{\Re[\mathcal{H}(\omega)]}\right) + \pi, & \text{otherwise.} \end{cases}\end{aligned} \tag{3.17}$$

Where $\Re(c)$ and $\Im(c)$ are the real and imaginary parts of the complex number c. The condition on the definition of $\phi(\mathcal{H}(\omega))$ is necessary because the range of the inverse tangent function is $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Without adding π to the function when $\Re[\mathcal{H}(\omega)] < 0$ we would not have a one to one correspondence between the complex form and the magnitude-phase form of $\mathcal{H}(\omega)$. Using these definitions we can rewrite $\mathcal{H}(\omega)$ in terms of its magnitude and phase:

$$\mathcal{H}(\omega) = |\mathcal{H}(\omega)|\mathrm{e}^{i\phi(\mathcal{H}(\omega))} \tag{3.18}$$

So an input to this filter with frequency ω would have its amplitude scaled by $|\mathcal{H}(\omega)|$ and its phase shifted by $\phi(\mathcal{H}(\omega))$ relative to the value at the mid-point.

The frequency response of an arbitrary filter with $2m+1$ coefficients is:

$$\begin{aligned}\mathcal{H}(\omega) &= \sum_{n=-m}^{m} c_n \mathrm{e}^{-i\omega n} \\ &= c_{-m}\mathrm{e}^{-i\omega(-m)} + c_{-m+1}\mathrm{e}^{-i\omega(-m+1)} + \cdots + c_0\mathrm{e}^{0} + \cdots + c_{m-1}\mathrm{e}^{-i\omega(m-1)} + c_m\mathrm{e}^{-i\omega m}\end{aligned} \tag{3.19}$$

If a symmetric filter is used, that is one where $c_n = c_{-n}$ for all n, then the response equation can be simplified:

$$\mathcal{H}(\omega) = c_m \left(\mathrm{e}^{i\omega m} + \mathrm{e}^{-i\omega m}\right) + c_{m-1} \left(\mathrm{e}^{i\omega(m-1)} + \mathrm{e}^{-i\omega(m-1)}\right) + \cdots + c_0 \tag{3.20}$$

and again, using Euler's formula, we can simplify this expression:

$$\mathcal{H}(\omega) = c_m.2\cos(\omega m) + c_{m-1}.2\cos(\omega(m-1)) + \cdots + c_0 \tag{3.21}$$

Equation 3.21 demonstrates that the frequency response of a filter with real symmetric coefficients is itself real. By equation 3.17 this means that the relative phase of the filter output and the data at the mid-point of the filter, is zero (or π) for all frequencies. In comparison with the mid-point, the output of the filter has had just the magnitudes of the frequencies altered and not the phases. If the filter has an even number of coefficients then its output will be in phase with the point half-way between the two central inputs. In the application presented in this thesis, odd order symmetric filters are used to easily ensure that the output is in phase with a known input point. This type of filter with a zero phase response at the centre is very important in many applications, particularly for speech and music processing and for data transmission, where a change in the relative phases of the frequencies would produce an unacceptable distortion.

3.3.2: Filter Design

When designing a digital filter we have in mind a particular frequency response that we would like it to possess. We would like to minimise the difference, $\mathcal{E}(\omega)$, between the desired frequency response, $\mathcal{D}(\omega)$, and the actual response, $\mathcal{H}(\omega)$, of the designed filter.

$$\mathcal{E}(\omega) = \mathcal{D}(\omega) - \mathcal{H}(\omega) \tag{3.22}$$

$\mathcal{D}(\omega)$, $\mathcal{H}(\omega)$ and $\mathcal{E}(\omega)$ are all complex numbers. We would like to weight the real and imaginary parts equally and so we minimise the absolute value of $\mathcal{E}(\omega)$. By also taking the square of $|\mathcal{E}(\omega)|$, larger errors are emphasised relative to smaller ones. As we are interested in the quality of the filter over all frequencies we will integrate from $-\pi$ to π to get the error function, $\mathcal{E}$:

$$\mathcal{E} = \int_{-\pi}^{\pi} |\mathcal{E}(\omega)|^2 \, d\omega \tag{3.23}$$

Minimising this error function with respect to all the filter coefficients, c_n, provides the Fourier filter design equation:

$$c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{D}(\omega) \mathrm{e}^{i\omega n} \, d\omega \tag{3.24}$$

A proof of this result is given in appendix A.3.

Each coefficient in the Fourier design method is calculated independently of the others. This is not usually the case in more advanced design methods. One result of the design equation 3.24 is that if the desired response function, $\mathcal{D}(\omega)$ is even then the coefficients, c_i, will be real:

$$\begin{aligned} c_n &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{D}(\omega) \mathrm{e}^{i\omega n} \, d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{D}(\omega) \cos \omega n \, d\omega + \frac{i}{2\pi} \int_{-\pi}^{\pi} \mathcal{D}(\omega) \sin \omega n \, d\omega \end{aligned} \tag{3.25}$$

Looking at the terms being integrated, in the first term we have the product of two even functions which yields an even function and in the second term we have the product of an even ($\mathcal{D}(\omega)$) and an odd function (sine) which yields an odd function. The integral of an odd function over a region centred at zero is zero. Equation 3.25 thus becomes:

$$\begin{aligned} c_n &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{D}(\omega) \cos \omega n \, d\omega \\ &= \frac{1}{\pi} \int_{0}^{\pi} \mathcal{D}(\omega) \cos \omega n \, d\omega \qquad \Rightarrow \quad \Im(c_n) = 0 \end{aligned} \tag{3.26}$$

If $\mathcal{D}(\omega)$ was not an even function, that is, if $\mathcal{D}(\omega) \neq \mathcal{D}(-\omega)$ for some ω, then given the discussion on negative frequencies in section 3.1, the design would not make much sense. The filters are therefore designed for the region 0 to π and it can be assumed that they are symmetrical about 0.

3.3.3: Filter Properties

If we have a filter of length $2m + 1$ with coefficients a_n, then we can create a new filter by scaling the filter coefficients. If we let $c_n = da_n$ for all n, where d is our scale factor,

then the frequency response of c_n will simply be the frequency response of a_n scaled by d:

$$\begin{aligned}\mathcal{H}_c(\omega) &= \sum_{n=-m}^{m} c_n \mathrm{e}^{-i\omega n} \\ &= \sum_{n=-m}^{m} d a_n \mathrm{e}^{-i\omega n} \\ &= d \sum_{n=-m}^{m} a_n \mathrm{e}^{-i\omega n} \\ &= d\mathcal{H}_a(\omega)\end{aligned} \tag{3.27}$$

A new filter may also be created by adding a constant to the middle coefficient. Doing so raises or lowers the magnitude of the whole of the frequency response by the constant. This can be seen from equation 3.15. Both these techniques are used in chapter 5 to quickly create new filters.

If we have two filters, a_n and b_n, then we can create new filters by taking the sum or the product of the corresponding coefficients in the two filters. Both filters need to have the same number of coefficients. We will assume that they both have infinite order. This is achieved by symmetrically by padding the filters with zeroes.

The result of taking the sum of the coefficients is easy to show:

$$\begin{aligned}\text{Let} \quad & c_n = a_n + b_n \quad \forall n \\ \text{Then} \quad & \mathcal{H}_c(\omega) = \sum_{n=-\infty}^{\infty} (a_n + b_n)\mathrm{e}^{-i\omega n}\end{aligned} \tag{3.28}$$

where $\mathcal{H}_c(\omega)$ is the frequency response for coefficients c_n. The simplification is then quite trivial:

$$\begin{aligned}\mathcal{H}_c(\omega) &= \sum_{n=-\infty}^{\infty} a_n \mathrm{e}^{-i\omega n} + \sum_{n=-\infty}^{\infty} b_n \mathrm{e}^{-i\omega n} \\ &= \mathcal{H}_a(\omega) + \mathcal{H}_b(\omega)\end{aligned} \tag{3.29}$$

So if the coefficients of a and b are added then the sum of their frequency responses is also taken.

If we let $c_n = a_n b_n$ then the result is much more complicated.

$$\begin{aligned} \mathcal{H}_c(\omega) &= \sum_{n=-\infty}^{\infty} c_n \mathrm{e}^{-i\omega n} \\ &= \sum_{n=-\infty}^{\infty} a_n b_n \mathrm{e}^{-i\omega n} \end{aligned} \tag{A.28}$$

The final equation is derived in appendix A.4 and presented here:

$$\mathcal{H}_c(\omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{H}_a(\omega - s)\mathcal{H}_b(s)\, ds \tag{3.31}$$

The frequency response of the coefficients c_n is the "periodic convolution" of the responses of coefficient sets a_n and b_n. To find $\mathcal{H}_c(\omega)$ we integrate across the product of the mirror image of $\mathcal{H}_a(s)$ shifted by ω, and $\mathcal{H}_b(s)$. This is equivalent to considering the cross-correlation between $\mathcal{H}_a(-s)$ and $\mathcal{H}_b(s)$, at each ω in the range $-\pi$ to π.

Figure 3.8 breaks down a convolution calculation into several steps. We are calculating $\mathcal{H}_c(\omega)$ in terms of $\mathcal{H}_a(\omega)$ and $\mathcal{H}_b(\omega)$ where $c_i = a_i b_i$. Figure 3.8(a) shows the idealised responses $\mathcal{H}_a(\omega)$ and $\mathcal{H}_b(\omega)$. In figures 3.8(b)–(g) the dummy variable has switched to s and $\mathcal{H}_a(\omega - s)$ and $\mathcal{H}_b(s)$ are plotted with a range of values for ω. The shaded areas correspond to the integral of the product of the two functions. Finally, in part (h) we take the integrated areas and divide by 2π to get $\mathcal{H}_c(\omega)$.

3.3.4: Windowing

The Fourier method was derived with no restriction placed on the number of coefficients. As a result, it only gives an exact solution if an infinite number of coefficients is used. In practice this is impossible. If, instead of calculating an infinite number of coefficients, we just calculate $2m + 1$ coefficients then the frequency response, $\mathcal{H}(\omega)$, will not match the desired response, $\mathcal{D}(\omega)$, perfectly. Figure 3.9(a-c) shows the desired response and the response resulting from truncation at 21, 41 and 201 coefficients. Unsurprisingly, the more coefficients that are used, the closer $\mathcal{H}(\omega)$ is to $\mathcal{D}(\omega)$.

We can think of the truncation of the set of coefficients c_n as the product of that set with a "uniform window", w_i:

$$\text{Let} \quad d_n = c_n w_n \quad \text{where} \quad w_n = \begin{cases} 1, & \text{if } |n| \leq m; \\ 0, & \text{otherwise} \end{cases} \tag{3.32}$$

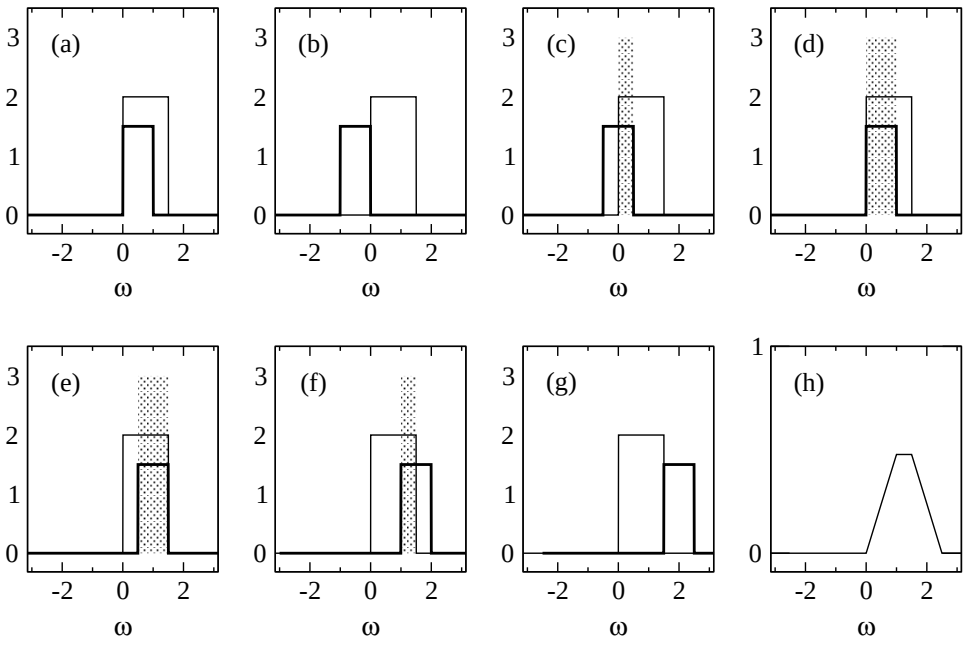

Figure 3.8. Convolution of $\mathcal{H}_a(\omega)$ (thick line) with $\mathcal{H}_b(\omega)$ (thin line): (a) $\mathcal{H}_a(\omega)$ and $\mathcal{H}_b(\omega)$; (b)–(g) $\mathcal{H}_a(\omega - s)$ and $\mathcal{H}_b(s)$ with ω taking the values π, $\pi + \frac{1}{2}$, $\pi + 1$, $\pi + 1\frac{1}{2}$, $\pi + 2$, $\pi + 2\frac{1}{2}$ respectively. The shaded regions correspond to the integral of the product of $\mathcal{H}_a(\omega - s)$ and $\mathcal{H}_b(s)$; (h) The complete convolution of the signals.

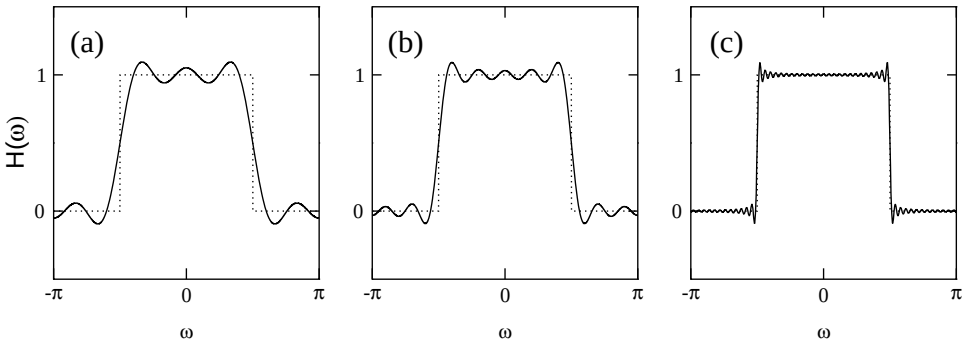

Figure 3.9. The desired frequency response (dotted line) and the actual response of the Fourier design method (solid line) for (a) 21 coefficients, (b) 41 coefficients and (c) 201 coefficients.

The frequency response of the truncated filter is then:

$$\mathcal{H}_d(\omega) = \sum_{n=-\infty}^{\infty} c_n w_n \mathrm{e}^{-i\omega n} \tag{3.33}$$

Which, from the convolution result obtained in section 3.3.3 can be rewritten as:

$$\mathcal{H}_d(\omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{H}_w(\omega - s)\mathcal{H}_c(s)\, ds \tag{3.34}$$

The top row of figure 3.10 reveals the origin of the ripples in $\mathcal{H}(\omega)$ visible in figure 3.9. In (a) the coefficients of the uniform window, w_i, are plotted. The window in this

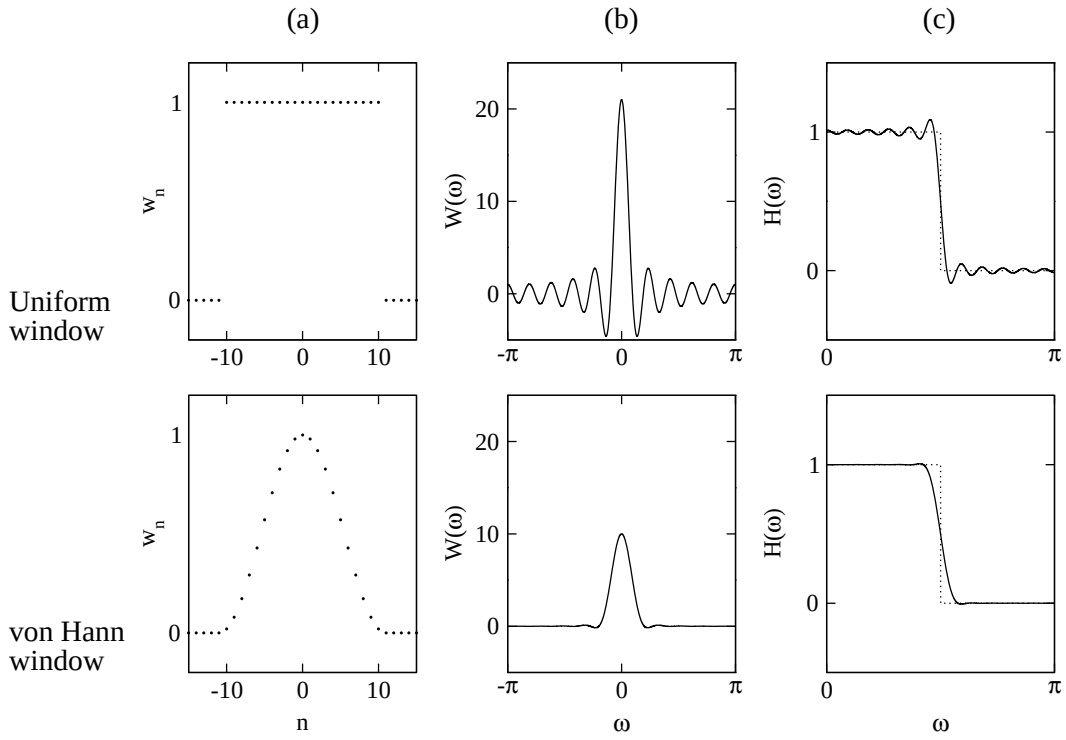

Figure 3.10. A comparison of the effects of the uniform and von Hann windows on the frequency response using the Fourier design method. Column (a) shows the window coefficients. Column (b) shows the response of the window. Column (c) is the convolution of the window response with the coefficients of the Fourier design method.

example has 21 coefficients at 1 and all the rest at 0. Part (b) shows the frequency response of the uniform window, $\mathcal{H}_w(\omega)$. It has a large peak in the middle and slowly diminishing ripples either side. The ideal frequency response generated by the infinite set of Fourier coefficients, $\mathcal{H}_c(\omega)$ is shown lightly in part (c). The bold line in part (c) is $\mathcal{H}_d(\omega)$ — the convolution of the ideal response with the window response. The effect of the window is to broaden the transition region and introduce ripples either side.

The uniform window is the most basic window of all and many other windows with differing properties are known. In the work presented in this thesis, the von Hann window (also known as the "Hanning window", "raised cosine window" or "cosine squared window") will be used in combination with the Fourier design method for much of the filter design.

Equation 3.35 gives the functional form of the von Hann window (where the length of the window is $2m+1$). The bottom row of figure 3.10 plots the window coefficients, their frequency response and the result of convolving the window response with the Fourier response. Looking at part 3.10(c) we can see that in comparison with the uniform

window, the transition region is much broader (i.e. it takes a larger frequency range for the response to change from 1 to 0) but the ripples either side of the transition are very much smaller. The width of the transition region can be reduced by making the filter longer. The von Hann window gives a good uniform response which is considered important in the application presented in chapter 4.

$$w_i = \frac{1}{2}\left(1 + \cos\left(\frac{\pi i}{m+1}\right)\right) \tag{3.35}$$

3.4: Summary

In any digital signal processing application, the sampling rate of the data is important. If the sampling rate is too low, then spurious signals will be generated. This is an important issue when deciding how frequently to store on disc, data being generated in an MD calculation.

The spectral density is a useful tool for analysing motion in an MD trajectory. It can be thought of as the simulation equivalent of an IR vibrational spectrum. The spectral density calculation employs a Fourier transform and must be interpreted with this in mind—the Fourier transform gives no information on the position in time of a frequency, only that it is present.

The digital filters used in this work are non-recursive. Their coefficients are real, symmetric and of odd order, which has the effect that if the filter output is compared with the mid-point of the filter input, then only the magnitudes of the frequencies are altered and the phases are untouched. The Fourier filter design method and the concept of windowing have also been described. This method of filter design will be used in chapter 4.

4: Digitally Filtered Molecular Dynamics

4.1: Prior Work

The roots of this project can be traced back to the application of frequency-domain filtering to MD analysis by Osguthorpe, Dauber-Osguthorpe *et al.*[73, 74, 76–79] The filtering was applied to the Cartesian coordinate trajectories of completed MD simulations. The technique involves three steps:

1. Fourier transforming each atomic trajectory (of length T steps) to the frequency domain:

$$H_n(\nu) = \frac{1}{T} \sum_{t=0}^{T-1} x_n(t) \mathrm{e}^{-2\pi i \nu t/T} \tag{4.1}$$

2. Applying a filtering function to remove frequencies that are not required:

$$\begin{aligned} F(\nu) &= \begin{cases} 1, & \text{if } \nu_{\min} < \nu < \nu_{\max}; \\ 0, & \text{otherwise.} \end{cases} \\ H'_n(\nu) &= H_n(\nu) F(\nu) \end{aligned} \tag{4.2}$$

3. Inverse Fourier transforming back to the time domain:

$$x'_n(t) = \sum_{\nu=0}^{T-1} H'_n(\nu) \mathrm{e}^{2\pi i \nu t/T} \tag{4.3}$$

The technique has been used to remove high-frequency motion to clarify the graphical display of a simulation trajectory and, in combination with the spectral density, for isolating and investigating the type of motions present in particular frequency bands. The drawback of frequency-domain filtering is that the Fourier transforms are expensive to compute (especially for long trajectories). Michael Levitt[80] took this idea of filtering the motion according to frequency and performed the filtering in the time-domain instead using non-recursive digital filters.

The purpose of Levitt's work was to remove high-frequency noise from the graphical display of pre-calculated simulation trajectories, to show more clearly the important low frequency motions. The method works in real-time and the frequency characteristics of the displayed trajectory can be altered interactively. To achieve this, a filter of $2m+1$ coefficients is used (m typically three), with each coefficient equal to $1/(2m+1)$. The inputs to the filter are the $3N$ atomic coordinate components (N being the number of atoms in the system). At each time-step, the filter is applied to the $3N$ signals separately, producing $3N$ filtered coordinates for display. The time-step of the displayed frame is taken as being the centre data point of the filter input.

A filter that weights each input identically may be called an averaging filter. One can see that by averaging atomic coordinates over a time period, high-frequency motion will cancel itself out and low-frequency motion will be relatively unaffected or indeed enhanced. A plot of the frequency response of this type of filter (with $m = 3$) is shown in figure 4.1. By taking m stored data points either side of the output position, Levitt ensured that the phase of the coordinates was unaffected or modified only by π. The frequency response can be changed by changing the length of the averaging filter. It can be shown[80] that if the time period the filter acts upon is t_f ps then all frequencies above approximately $20/t_f\,\mathrm{cm}^{-1}$ are attenuated by 0.5 or more. This cut-off is lowered by increasing the length of the filter.

Dauber-Osguthorpe *et al.*[56] then took the idea of a time-domain filter and applied it to the atomic velocity components of an evolving MD simulation. The method is called selectively enhanced molecular dynamics (SEMD) (see also section 2.2.3). The only published work using SEMD has been its application to hexane. Unfortunately, the description of the method in the paper is not complete so some assumptions have

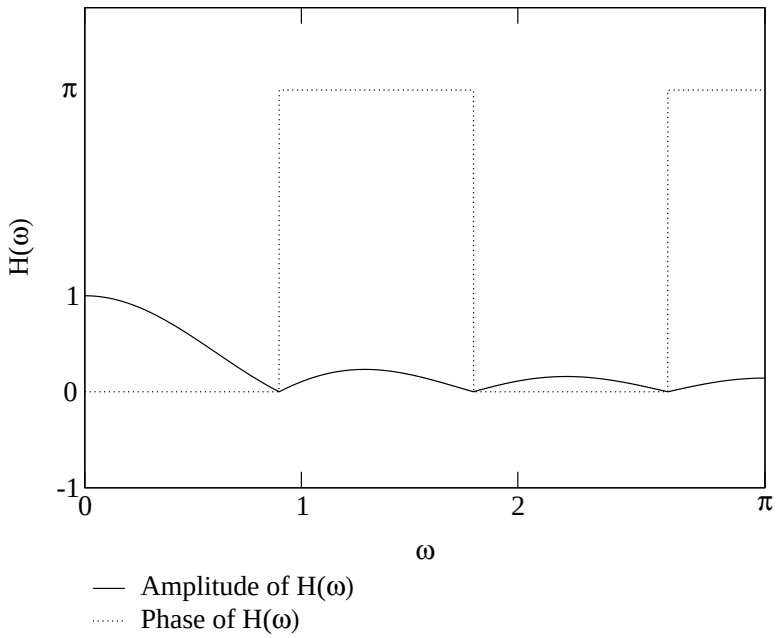

Figure 4.1. Frequency response of a seven coefficient digital filter with all coefficients equal to 1/7 as used by Levitt. The filter output has been compared to the middle input.

had to be made in the following description.

During the simulation, a buffer containing the previous 70 fs of atomic velocities is updated every time step. The filter has all coefficients equal to one and is applied to each of the $3N$ velocity-component signals from this buffer to produce a new set of atomic velocities. The new set replaces the velocities from the current time step and the MD can continue. As the simulation was performed at 300 K, we must also assume that the atomic velocities output by the filter are scaled such that the instantaneous temperature of the system immediately before and after the filter application is kept constant.

If we assume that a 1 fs time-step was used, then the filter must have 70 coefficients. Figure 4.2 shows the frequency response of this filter. The amplification of the low frequency region relative to the high frequency region is 70—a massive difference. In aligning the filter output with the final input, the phase response of the filter is no longer very useful. Not only are the different frequency atomic-velocities out of phase with each other, but also the atomic velocities will be out of phase with the atomic coordinates. This leads to uncontrolled and indeed undesirable behaviour. The relationship between the phase of $\mathcal{H}(\omega)$ and the frequency, ω, is essentially linear but the phase plot jumps by

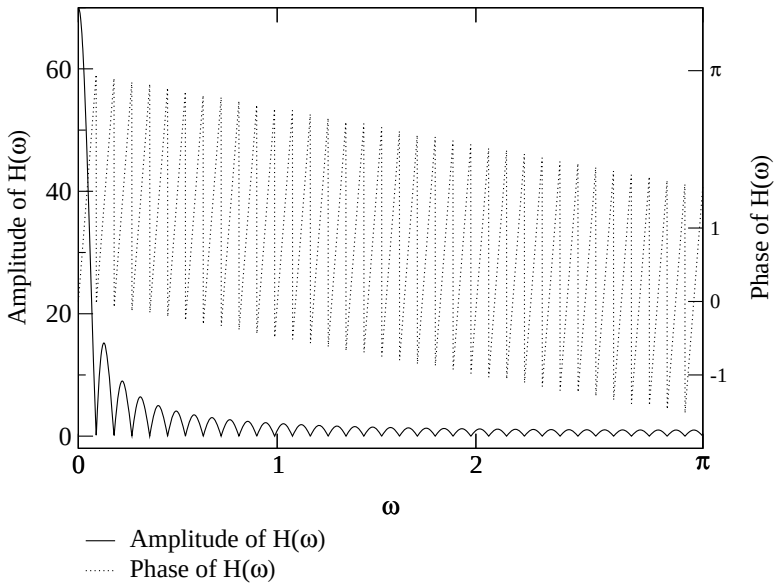

Figure 4.2. Frequency response of a seventy coefficient digital filter with all coefficients equal to 1. This is the filter that Dauber-Osguthorpe used in the SEMD method (assuming a time step of 1 fs). The filter output has been compared to the final input.

π every time the amplitude plot reaches the x-axis. This is because the representation of the amplitude response has been kept positive so a negative amplitude response has been represented by its magnitude and a π shift in the phase.

Two SEMD simulations were reported in the paper: a simulation where the filter was applied continuously (cSEMD) and a simulation with an intermittent filter application (iSEMD) where the filtering was carried out in 1 ps intervals alternating with 0.5 ps normal MD calculations. The prospect of applying the filter, with its uncontrolled phase character, for 1000 steps in a row does not seem good, as all but the first filter application will be filtering a buffer containing phase discontinuities and ill-behaved waveforms.

The first stage of this project was the implementation of the cSEMD technique. DL_POLY (ref. 18) was chosen as a framework for the SEMD code because DL_POLY is clearly written and well documented. The method as described above was implemented and a model of hexane was defined and simulated. At this stage, another problem with cSEMD was noticed. When the filter is applied, some energy can be transferred into rotational degrees of freedom. Once energy is in the rotational degrees of freedom it has no way of being transferred back again, and the rotation just builds up until the molecule

is spinning rapidly but not vibrating at all. To counter this, a step that removed the rotation while keeping the instantaneous temperature constant was introduced using the following protocol:

1. Calculate the temperature of the system, T_{initial}.
2. Calculate the centre of mass (COM) coordinate and velocity.
3. Translate the COM to the origin and zero the translational velocity.
4. Calculate the angular momentum of each atom about the origin and sum these to give the total angular momentum of the rigid body.
5. Using the inverse of the inertial tensor, calculate the angular velocity of the rigid body.
6. Translate the coordinates back to the original origin.
7. Remove from each atom the velocity due to the angular velocity.
8. Calculate the temperature, T_{final}, and scale the velocities by $(T_{\text{initial}}/T_{\text{final}})^{1/2}$.

When this modification was used, the conformational change produced was very similar to that observed by Dauber-Osguthorpe *et al.*[56] Figure 4.3 plots the three dihedral angles in hexane over a 100 ps cSEMD simulation. Unfortunately, although the simulation is now sampling the conformations of hexane, the system is no longer stable. Figure 4.4 displays the simulation temperature as a function of time. Clearly the application of a filter in this way is completely destabilising the rest of the molecular dynamics calculation.

4.2: Theory

The digitally filtered molecular dynamics (DFMD) method has been developed to address the two perceived flaws in the SEMD method, namely the lack of precise control over the frequency response of the filter, and the instability caused by the filtered velocities being out of phase with the rest of the simulation state.

In DFMD, as in SEMD, the atomic velocity components are filtered. Filter coefficients are chosen (using a design method such as those discussed in sections 3.3.2 and 3.3.4) to approximate a desired frequency response which may target selected frequency motion for enhancement or suppression. The filter type used is a non-recursive, real, symmetric filter, with $2m + 1$ coefficients (see section 3.3 and ref. 81). These designed

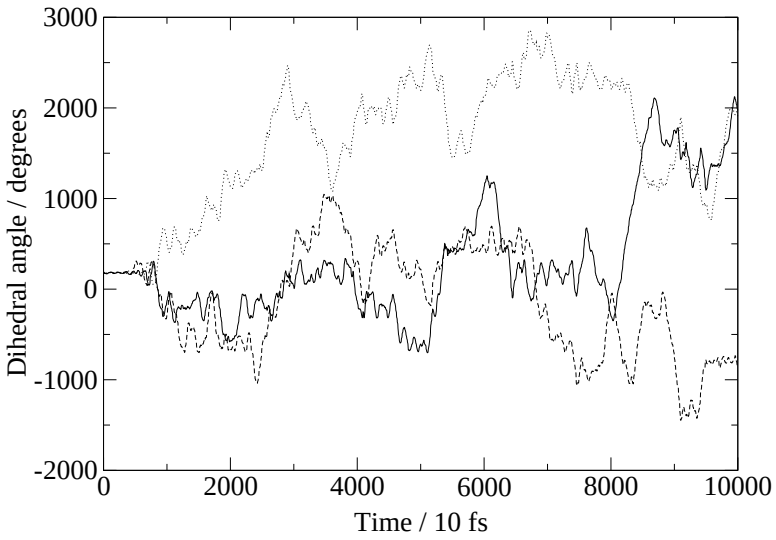

Figure 4.3. cSEMD simulation of hexane: the three dihedral angles are plotted.

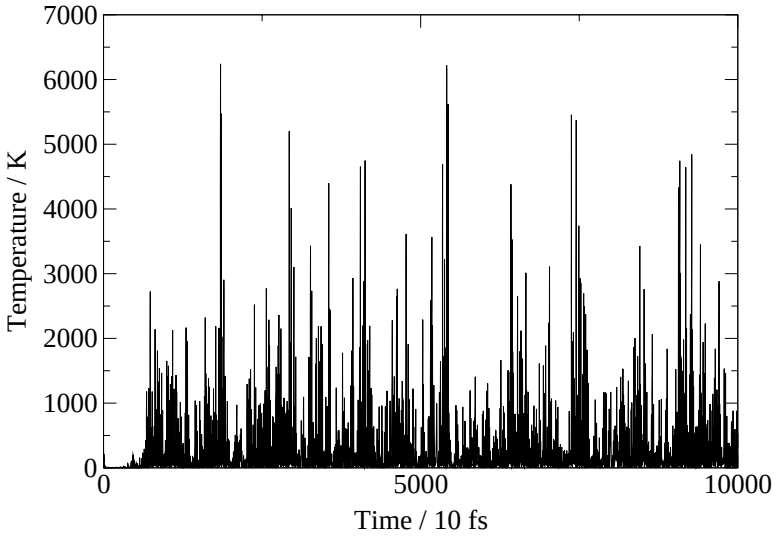

Figure 4.4. cSEMD simulation of hexane: the temperature during the "NVT" simulation.

filters provide the required precision in the control of the frequencies.

In the first instance, normal MD was performed for $2m+1$ time-steps, saving all the velocities at each step and saving the system state (coordinates and forces) at the middle step $(m+1)$. The filter was then separately applied to the x, y and z components of each of the atomic velocity trajectories to produce a new filtered set of atomic velocities. The filtered velocity set was then combined with the saved set of coordinates and velocities and normal MD restarted from the middle step. Apart from using designed filters, this protocol is significantly different from the SEMD method in a further two ways—the

filter is applied only once, and the filtered velocities are combined with the forces and coordinates from the centre step of the filter buffer. The output of a real, symmetric filter is in phase with the centre point of the filter buffer (see equations 3.19–3.21 and 3.17) and so by combining the filtered velocities with the system state from the mid-point we avoid the phase problems apparent in the SEMD method.

It quickly became apparent that a single application of the filter was not sufficient. Sometimes the filter was effective and sometimes it was not, depending on what point in the trajectory it was applied. The timing of the filter pulse is critical. Let us suppose we have a diatomic molecule vibrating at $1600\,\mathrm{cm}^{-1}$ and a filter composed entirely of zeros. The frequency response of the filter will be zero for all frequencies. Applying the filter to the diatomic will certainly zero the velocities of the atoms. However, if the length of the bond at the time of the filter application was not equal to the equilibrium bond length then the atoms would start moving again due to the forces acting upon them. If the filter was applied at an extreme of the motion (when the velocities were already zero) then it would have no effect at all on the system dynamics. The filter acts only upon the velocities and hence the kinetic energy; it has no effect on the coordinates and potential energy of the system.

There is an elegant solution to this problem: two filter-pulses are used, $\pi/2$ radians apart (figure 4.5(a)). If $\nu\,\mathrm{Hz}$ is the target frequency and $\Delta t\,\mathrm{s}$ is the time-step then the number of steps delay needed between filter-pulses, D, is given by equation 4.4.

$$D = \frac{1}{4\nu\Delta t} \tag{4.4}$$

To use two filter-pulses the stored velocity-set needs to be extended by $\pi/2$ radians (D time-steps). The first filter application uses $2m+1$ time-steps from the beginning of the buffer (the top grey bar in figure 4.5) and the second application uses $2m+1$ stored time-steps from the end of the buffer (the bottom grey bar). Normal MD is performed until the $2m+1+D$ buffer of velocities is filled. The first pulse is then applied and the filtered velocity set is combined with the system state from position $m+1$ in the buffer. The system is then allowed to evolve using normal MD for a further D time-steps and the second filter pulse is then applied. The second set of filtered velocities is combined

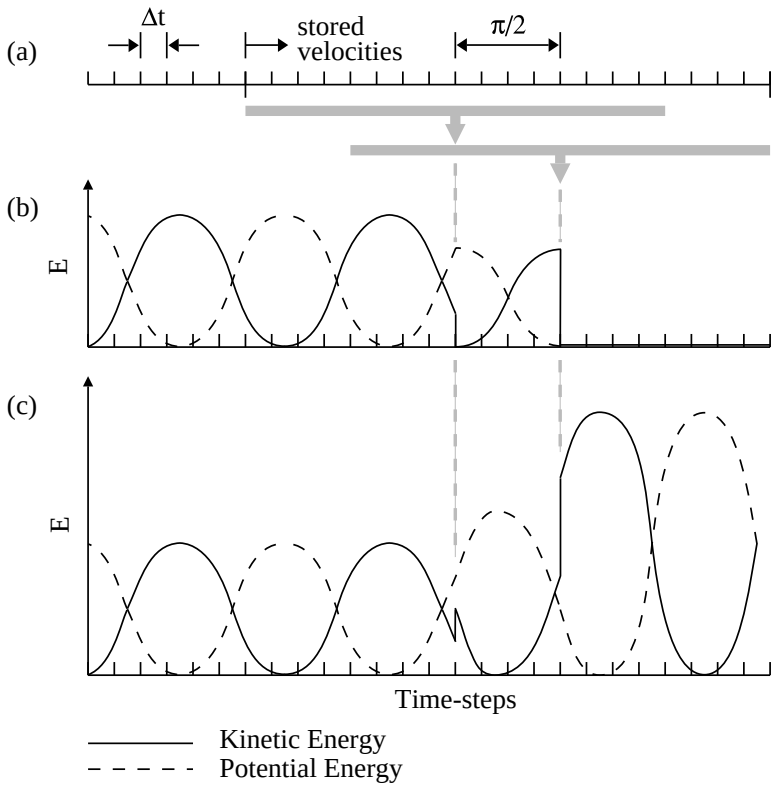

Figure 4.5. Illustration of the double-pulse technique for velocity suppression and amplification: (a) the progression of time with two grey bars showing the two sets of velocities used by the filter, (b) the effect of a total suppression filter on the KE and PE, (c) the effect of a $\sqrt{2}$ magnitude velocity-amplifying filter on the KE and PE.

with the system state at the current time-step and the simulation continues with normal MD.

In figure 4.5(b) the kinetic and potential energies during a simulation in which the frequency is quashed is shown. The initial filter-pulse zeros the velocities using the first $2m+1$ data points from the store. The system is then left to evolve for $\pi/2$ radians, during which time the kinetic energy increases to its new maximum. A second filter pulse, using the last $2m+1$ data points, is then applied which again zeros the velocities. This time both the kinetic and potential energies are zero and the frequency is quashed. The case of frequency amplification is shown in figure 4.5(c). The first pulse is applied and $\pi/2$ radians after the first pulse, the extra energy imparted into the system has been transfered into potential energy. The second filter pulse imparts enough kinetic energy to achieve full amplification.

The two-pulse technique will work exactly in an ideal system. If we have a harmonic oscillator of frequency $\omega\,\mathrm{rad\,s^{-1}}$ and phase $\delta\,\mathrm{rad}$ then its kinetic and potential energies will also be harmonic but of frequency $2\omega\,\mathrm{rad\,s^{-1}}$. Let the kinetic energy at time t be $T(t)$ and the potential energy be $U(t)$. The method relies on one mathematical result: that $U(t+\Delta t) = T(t)$ where $\Delta t = \pi/2\omega$ is the time needed for half an energy cycle (D time-steps). If E is the total energy then we have:

$$\begin{aligned} E &= T(t) + U(t) \\ T(t) &= \frac{1}{2}E(1+\cos(2\omega t - 2\delta)) \\ U(t) &= \frac{1}{2}E(1-\cos(2\omega t - 2\delta)) \\ U(t+\Delta t) &= \frac{1}{2}E(1-\cos(2\omega(t+\Delta t) - 2\delta)) \\ &= \frac{1}{2}E(1-\cos(2\omega t - 2\delta + \pi)) \\ &= \frac{1}{2}E(1+\cos(2\omega t - 2\delta)) \\ &= T(t) \end{aligned} \tag{4.5}$$

It can similarly be shown that $T(t+\Delta t) = U(t)$.

The amplification, A, of the kinetic energy at frequency ω will be $A = [\mathcal{H}(\omega)]^2$ since we are amplifying (or suppressing) the velocities and the kinetic energy is a function of the square of the velocities. With the first pulse at time $t = t_0$ the kinetic energy goes from $T(t_0)$ to $T_1(t_0)$ where T_1 is the kinetic energy of the wave after one filter pulse:

$$T_1(t_0) = AT(t_0) \tag{4.6}$$

D time-steps later, the potential and kinetic energies will have exchanged so that $U_1(t_0+\Delta t) = T_1(t_0)$. The second filter pulse (acting on the stored buffer) combines these coordinates with a new velocity set and a new kinetic energy, T_2:

$$\begin{aligned} T_2(t_0+\Delta t) &= AT(t_0+\Delta t) \\ &= AU(t_0) \end{aligned} \tag{4.7}$$

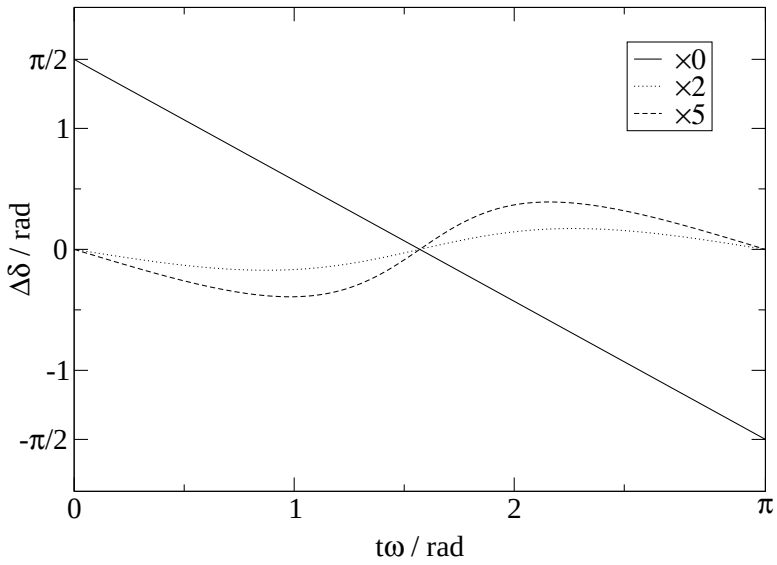

Figure 4.6. The phase change, $\Delta\delta$, in the oscillation on the application of the first filter pulse as a function of application time.

The total energy at time $t + \Delta t$ is then:

$$\begin{aligned} E_2 &= T_2(t_0 + \Delta t) + U_1(t_0 + \Delta t) \\ &= AU(t_0) + T_1(t_0) \\ &= AU(t_0) + AT(t_0) \\ &= AE \end{aligned} \tag{4.8}$$

This demonstrates that for the case of a simple harmonic oscillator, the two pulses result in the correct amplification.

Although the filter itself produces no change in the phase of the velocities of the targeted frequencies with respect to the original velocities, the propagation of the trajectory using the equations of motion necessarily produces a small phase change. This phase change is evident in figure 4.5(b)—at both points of application, the kinetic energy oscillation is shifted in phase such that it is at a minimum. It may be shown (see appendix A.5) that the change in phase, $\Delta\delta$, of the targeted frequency is:

$$\Delta\delta = \frac{1}{2}\left(2\omega t_0 - \cos^{-1}\left[\frac{(A-1) + (A+1)\cos(2t_0\omega - 2\delta)}{(A+1) + (A-1)\cos(2t_0\omega - 2\delta)}\right]\right) - \delta \tag{4.9}$$

Figure 4.6 plots this phase change across the whole range of different application times. One can see that even for the large amplification of $\times 5$, the change in phase has

a maximum of $0.392 \approx \pi/8$ rad. The case of a filter that zeroes the target frequency is an extreme. Clearly, in abruptly bringing the kinetic energy to zero, the phase of that energy oscillation can be shifted by as much as π and, equivalently, the phase of the atomic oscillation by $\pi/2$. Figure 4.6 shows that for this filter the phase shift varies linearly between $\pi/2$ and $-\pi/2$. Even this worst case is not very bad as the $\pi/2$ shift will only be present until the application of the second filter pulse, when the rest of the energy in that frequency will be removed. Generally this quenching filter has only been used for testing purposes; most filters used in this thesis are small amplifying filters.

The application of the second filter pulse brings the motion in the target frequency back into phase with the original wave. This can easily be shown using an energy argument and the identity given in equation 4.5:

$$\begin{aligned} T_2(t_0 + \Delta t) &= AT(t_0 + \Delta t) \\ &= AU(t_0) \\ U_2(t_0 + \Delta t) &= U_1(t_0 + \Delta t) \\ &= T_1(t_0) \\ &= AT(t_0) \end{aligned} \tag{4.10}$$

Using the same identity we can examine what the kinetic and potential energies of the original wave at time $t_0 + \Delta t$ would have been:

$$\begin{aligned} T(t_0 + \Delta t) &= U(t_0) \\ U(t_0 + \Delta t) &= T(t_0) \end{aligned} \tag{4.11}$$

So, comparing equations 4.10 and 4.11 we can see that after the second pulse, both the kinetic and potential energies have just been amplified by A. As they are in the same proportion, the phase shift is zero.

To summarise, for an isolated frequency, the second pulse of the filter produces the exact amplification with no phase change relative to the original wave. Between the two pulses there is generally a small phase change arising from the MD. If the target frequency was coupled to other oscillations (as is generally the case in an MD simulation) then the small phase change between the two pulses would have a slight effect on the motion, causing the second filter pulse to amplify a little imprecisely. However, this

combination of a zero phase-shift filter and a double pulse application clearly causes the least unwanted disruption.

If more than one frequency is being targeted then additional calculated pauses and filter-pulses are used and the length of the stored data-set is extended accordingly. Each targeted frequency requires a pair of filter-pulses separated by a calculated pause with normal MD in between. If, for two target frequencies, the pauses were 10 and 15 time-steps, then an initial filter-pulse would be used followed by a 10 time-step delay and a second filter-pulse. The second filter-pulse can then be thought of as the first pulse of a new pair—a pause of 15 time-steps follows and a third filter-pulse finishes the sequence.

The application of the DFMD filters is not computationally demanding. However, it could be considered that the requirement to discard the final m steps of the stored trajectory is an extra computational expense. There exist design methods where the phase of the frequency response is explicitly specified along with the magnitude. Using one of these methods one could specify that the phase shift at the end of the buffer was zero and so the last m steps of the trajectory would not have to be discarded. However, in attempting to control the phase of the response as well as the magnitude, the precision of the filter is reduced in comparison to the magnitude-only methods (for the same number of coefficients) and so to get an equivalent magnitude-response more coefficients would be needed. Moreover, however many coefficients are used, these filters cannot provide the mathematically perfect zero phase response of the centre point of a real symmetric filter used in this work. In reality, the final m steps are not really being "discarded" as that part of the trajectory may be analysed and plays a full part in the outcome of the filtered trajectory.

4.3: Applications

In this section, four applications are considered: a single, fully-flexible water molecule, a single molecule of butane, liquid butane, and the Syrian hamster prion protein. The first three simulations were performed in the NVE micro-canonical ensemble using a modified version of DL_POLY. In the gas-phase simulations the velocities were selected from a Gaussian distribution using the standard DL_POLY routine to give an initial temperature of 300 K. Normal mode analyses were performed using BOSS (ref. 6).

FOCUS (ref. 77) was initially used to calculate the spectral densities but in order to obtain the required control over the calculation, a program using Fourier transform routines from FFTW (ref. 82) was written in C. To design the filters, a program implementing the Fourier design method and von Hann window detailed in sections 3.3.2 and 3.3.4 was written in Perl.

4.3.1: Water

This system was chosen for study because there are only three normal modes. The frequencies of the stretching modes and the bending mode are widely spaced. This permits a detailed and clear investigation of selective frequency enhancement and suppression. In these simulations a time-step of 0.1 fs was used. This is much smaller than is generally recommended but was chosen to ensure complete energy conservation and to provide detailed data on the workings of DFMD. The force field parameters were taken from Teleman's flexible SPC model.[83]

The water molecule vibrates at three frequencies. The symmetric and asymmetric stretches are almost degenerate, as can be seen from the normal mode analysis given in figure 4.7(a). The spectral density obtained from a 1 ps MD simulation is also shown. The vibrational frequencies are centred at approximately $1519\,\mathrm{cm}^{-1}$ and $3716\,\mathrm{cm}^{-1}$. Four applications of DFMD present themselves: quenching the bending motion, enhancing the bending motion, quenching the bond stretches and enhancing the bond stretches. Each of these is examined in turn.

To quench the motion at $1519\,\mathrm{cm}^{-1}$ a 4001 coefficient filter was designed. The filter has a broad passband centred at $3716\,\mathrm{cm}^{-1}$ giving $\mathcal{H}(1519) = 0$ and $\mathcal{H}(3716) = 1$. Since the motion being targeted is at $1519\,\mathrm{cm}^{-1}$, the filter is applied as a double pulse with a pause of 55 time-steps in between. As stated in section 4.2, this corresponds to a $\pi/2$ time-delay (see equation 4.4). The frequency response of the digital filter together with the spectral density obtained after its application are depicted in figure 4.7(b). In comparison with figure 4.7(a) it is clear that the $1519\,\mathrm{cm}^{-1}$ motion has ceased but the $3716\,\mathrm{cm}^{-1}$ vibrations remain. The effect of the filter on the atomic motion is demonstrated in figure 4.8. In this figure and all others that show some property as a function of time, the section of the simulation that is over-written after the filter application is

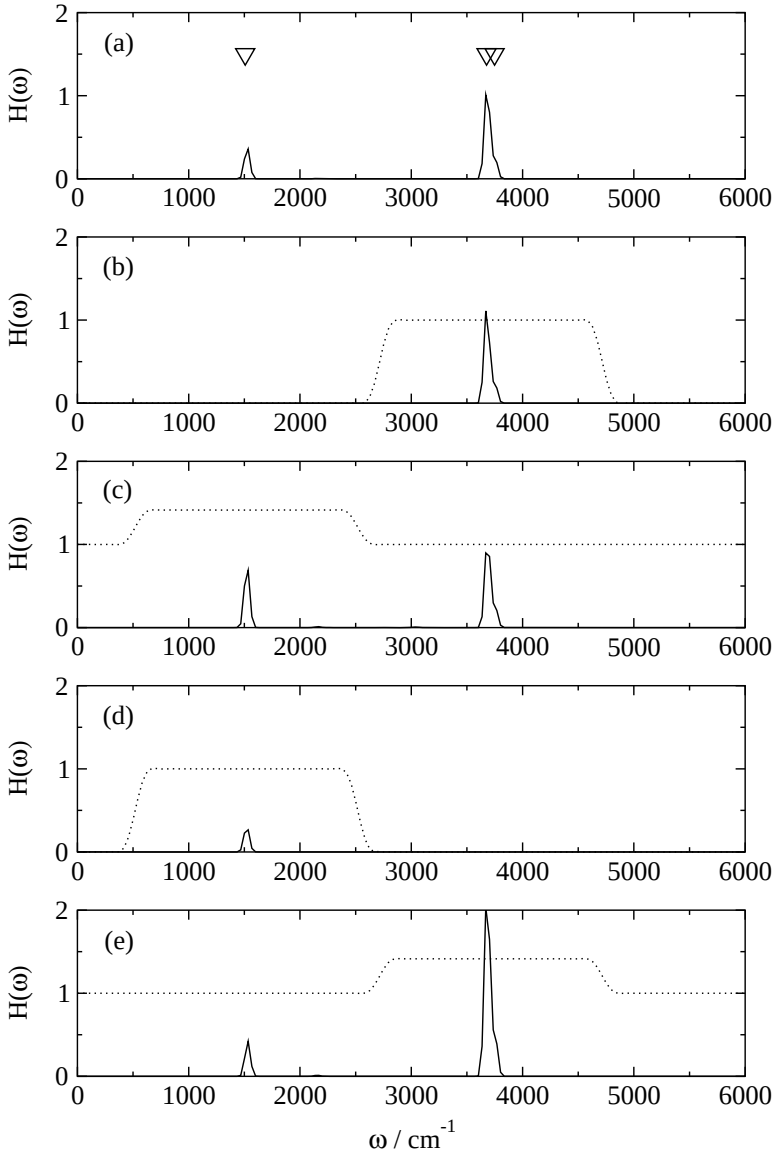

Figure 4.7. Water: (a) spectral density and NM analysis, (b)–(e) filters and resulting spectral densities. The normal mode frequencies are marked by triangles.

not displayed. Figure 4.8 confirms the analysis of the frequency distribution function—clearly the bending motion has largely ceased and the bond stretching has continued fairly unaffected. The remaining bending motion is at the higher frequency of 3716 cm^{-1}. Figure 4.9(a) shows how the stretching energy, bending energy and total energy varies during the simulation. When the 1519 cm^{-1} motion is quenched, the total energy drops by 2.7 $\mathrm{kJ\,mol}^{-1}$—slightly less than the total bending energy in the system. The bond stretching motion, predominantly at 3716 cm^{-1}, has a small 1519 cm^{-1} component. The 1519 cm^{-1} quenching filter has therefore removed energy from both the bending and the

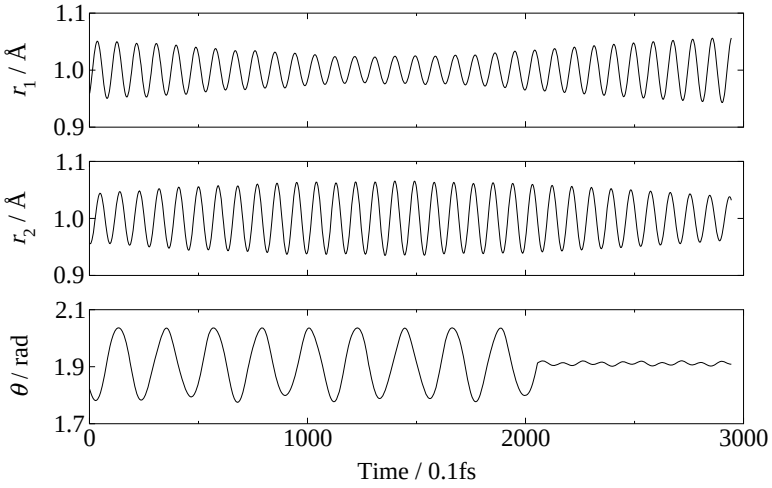

Figure 4.8. Water bond lengths r_1 and r_2 and bond angle θ before and after application of a $1519\,\mathrm{cm}^{-1}$ suppressing filter at step 2001.

stretching vibrations. Further evidence for the stretching and bending being affected by both frequencies is provided by figures 4.9(a) and (c): in (a) when the $1519\,\mathrm{cm}^{-1}$ motion is removed, the energy plot of the stretching motion is much more regular; similarly, in (c) when the $3716\,\mathrm{cm}^{-1}$ motion ceases, the bending motion becomes more regular.

To enhance the motion at $1519\,\mathrm{cm}^{-1}$, the filter depicted in 4.7(c) was used. This 4001 coefficient filter has $\mathcal{H}(1519) = 1.4142 \approx \sqrt{2}$ and $\mathcal{H}(3716) = 1$. As with the $1519\,\mathrm{cm}^{-1}$ suppression simulation, a double pulse separated by 55 time-steps was used. The spectral density of the simulation following the filter application is also shown in figure 4.7(c)—the peak at $3716\,\mathrm{cm}^{-1}$ is unchanged while the peak at $1519\,\mathrm{cm}^{-1}$ has increased in intensity. By amplifying the velocity components associated with the bend by 1.4142, the bending energy should be doubled (since $\mathrm{KE} = \frac{1}{2}mv^2$), this is indeed the case. Figure 4.9(b) demonstrates that the total system energy increases by $2.7\,\mathrm{kJ\,mol}^{-1}$.

To affect the motion at $3716\,\mathrm{cm}^{-1}$, a double pulse separated by 23 time-steps is required (see equation 4.4). Two 4001 coefficient filters were designed: one to suppress the motion and the other to enhance the velocities by 1.4142. The filters and the post-filter spectral densities are shown in figures 4.7(d) and (e). Once again the spectral densities demonstrate the near-perfect selectivity and control of DFMD. Looking at figures 4.9(c) and (d) we see that when the motion at $3716\,\mathrm{cm}^{-1}$ is quenched, the system energy drops by $11.3\,\mathrm{kJ\,mol}^{-1}$ and when the enhancement filter is applied it increases by $11.4\,\mathrm{kJ\,mol}^{-1}$. The total system energy is $13.7\,\mathrm{kJ\,mol}^{-1}$ and the sum of the energy

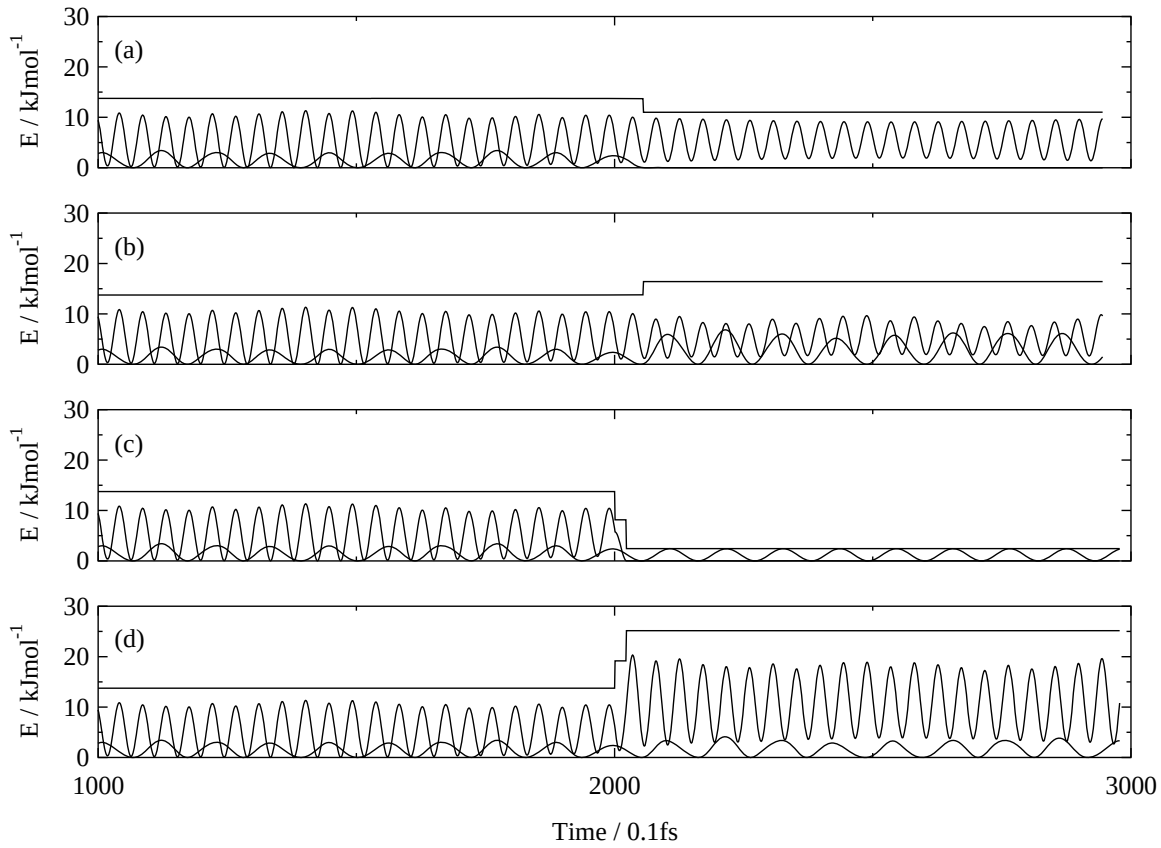

Figure 4.9. The total energy (top line), bond energy (large oscillation) and bend energy (small oscillation) of the water molecule before and after application of each of the four filters: (a) $1519\,\mathrm{cm}^{-1}$ suppressed, (b) $1519\,\mathrm{cm}^{-1}$ enhanced, (c) $3716\,\mathrm{cm}^{-1}$ suppressed, (d) $3716\,\mathrm{cm}^{-1}$ enhanced.

losses in the quenching experiments $(2.7 + 11.3)$ is $14.0\,\mathrm{kJ\,mol}^{-1}$. The values will never be in complete agreement because of the slight imprecisions inherent in the designed filters. When the simulations were run with 12001 coefficient filters closer agreement $(0.1\,\mathrm{kJ\,mol}^{-1})$ was achieved.

4.3.2: Gas-Phase Butane

Two sets of simulations of a single butane molecule were performed. First, detailed simulations using a 0.1 fs time-step were carried out on a fully flexible butane molecule. The bond stretching motions and the bond angle and torsional motions were targeted separately. Second, the torsional motion was enhanced in simulations using a 2 fs time-step and constrained bonds.

In the first set of simulations, the 1-2 and 1-3 interactions were taken from the standard OPLS united-atom force-field.[84] The 1-4 interaction uses a triple cosine potential

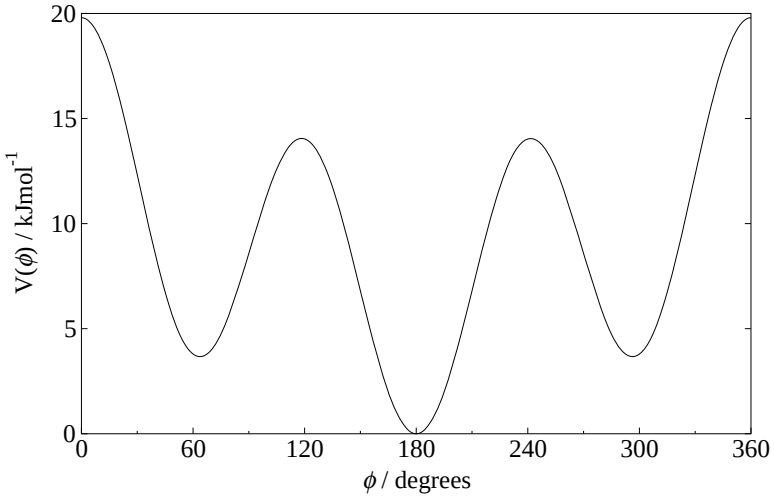

Figure 4.10. Potential energy function for the butane torsion.

with Fourier coefficients fitted to MM2 calculations.[85] The potential function for rotation about the central C-C bond is shown in figure 4.10. Using a time-step of 0.1 fs, a 5 ps simulation of unconstrained butane was run. The resulting spectral density is shown in figure 4.11(a) along with the results of a normal-mode analysis. The peak at $1047\,\mathrm{cm}^{-1}$ and the two overlapping peaks at $912\,\mathrm{cm}^{-1}$ are caused predominantly by bond stretching motions, the lowest frequency peak at $155\,\mathrm{cm}^{-1}$ is mainly due to torsional motion and the remaining two peaks at $296\,\mathrm{cm}^{-1}$ and $366\,\mathrm{cm}^{-1}$ are largely caused by bending motions. As before, two quenching and two enhancement experiments were run. The 4001 coefficient filters targeted either the three high-frequency (bond stretching) peaks or the three low-frequency (angle and torsion) peaks. To target two peaks with one filter requires a three-pulse application. The bond stretching peaks at $912\,\mathrm{cm}^{-1}$ and $1047\,\mathrm{cm}^{-1}$ require pulse delays of 91 and 80 time-steps; the bending and torsion peaks at $155\,\mathrm{cm}^{-1}$, $296\,\mathrm{cm}^{-1}$ and $366\,\mathrm{cm}^{-1}$ need a four-pulse application with delays of 538, 282 and 228 time-steps (see equation 4.4). Figures 4.11(b-e) show the filter frequency responses and the spectral densities resulting from their application. Since the bond stretching motion is so different in frequency to the bending and torsional motion, the excellent selectivity demonstrated in 4.11(b-e) can be achieved.

To selectively enhance conformational change, the torsional motion must be targeted. To affect only the peak at $155\,\mathrm{cm}^{-1}$ requires a filter with a narrow enhancement region. Such a precise filter requires either a large number of coefficients or a large time-

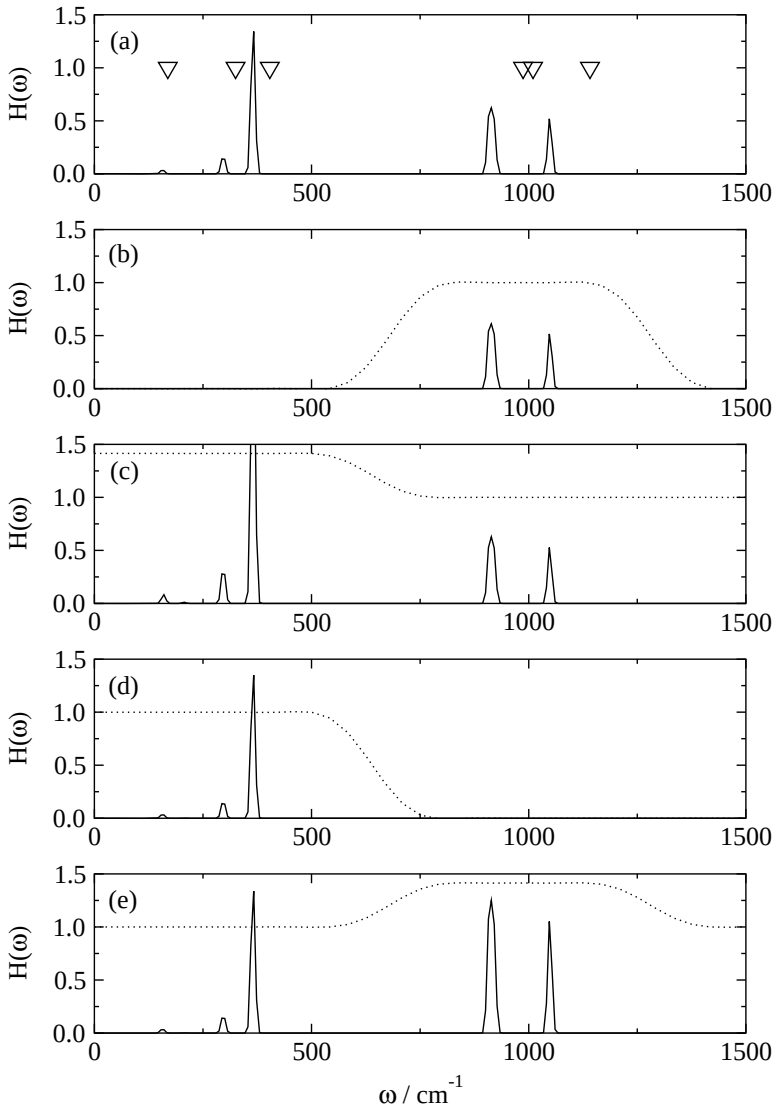

Figure 4.11. Butane: (a) spectral density and NM analysis, (b)–(e) filters and resulting spectral densities. The normal mode frequencies are marked by triangles.

step. In the following simulations, relatively small filters of 4001 coefficients along with a 2 fs time-step were used. The SHAKE (ref. 8) algorithm was applied to constrain the bond lengths. 1-4 non-bonded interactions were added using Lennard-Jones parameters from the OPLS united-atom force-field[85] to allow coupling between the angle bending motion and the torsional motion. The Fourier coefficients of the torsional potential were then re-parametrized to reproduce the curve given in figure 4.10.

The 4001 coefficient velocity-doubling filter shown in figure 4.12(a) was applied as the usual double pulse (27 time-step delay) three times to pump energy into the tor-

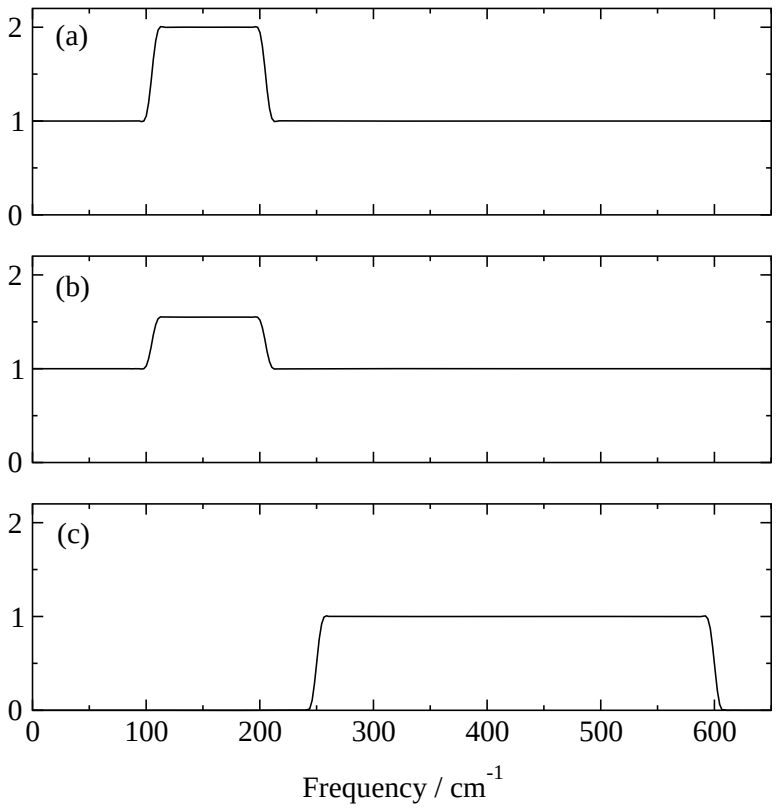

Figure 4.12. Frequency response of filters used to amplify and suppress the butane torsional motion.

sional motion. 4028 time-steps of velocity data were stored, the filter was applied as a double pulse and the simulation was allowed to continue while the velocity-store filled up again ready for the next pair of pulses. The maximum angular displacement of the dihedral angle from the trans conformation, $\phi_{\max}$, initially and after each filter application is listed in table 4.1. After the third filter application, the dihedral angle varies by $\pm 29.5°$ corresponding to a dihedral potential energy of approximately $6.6\,\mathrm{kJ\,mol^{-1}}$ (figure 4.10). The torsion only needs the lesser amplification given by the filter shown in figure 4.12(b) (which amplifies the velocities by 1.55) to push it over the barrier into the gauche conformation. A sample of the dihedral angle trajectory and the spectral density after these filters have been applied is shown in figure 4.13. The spectral density is now quite complicated. This is because the motion is profoundly anharmonic and both the gauche and the trans conformations are now being sampled. The molecule predominantly samples the gauche conformation but has not been given enough energy to cross the 0° barrier.

Table 4.1. Maximum dihedral angle displacement of butane, $\phi_{\max}$, during the enhancement process.

Stage	$\phi_{\max}$
Before filtering	3.89°
Velocities ×2	7.43°
Velocities ×2	14.37°
Velocities ×2	29.54°
Velocities ×1.55	153.04°

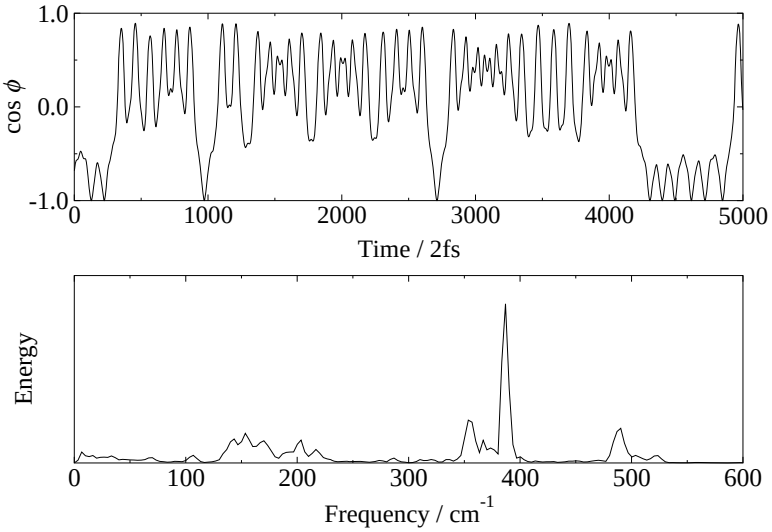

Figure 4.13. Dihedral angle trajectory and spectral density of butane after torsional amplification. The normal mode frequencies are marked by triangles.

As well as enhancing the torsional motion it is possible to quench it. Now that the molecule is oscillating between the trans and gauche conformations can we stop it oscillating and leave it in the gauche form? Looking at the spectral density (figure 4.13) it is clear that the dihedral angle is no longer vibrating with just one distinct frequency. The more predominant peaks at $143\,\mathrm{cm}^{-1}$, $153\,\mathrm{cm}^{-1}$, $169\,\mathrm{cm}^{-1}$, $203\,\mathrm{cm}^{-1}$, and $317\,\mathrm{cm}^{-1}$ will need to be quenched. The 4001 coefficient filter shown in figure 4.12(c) was designed to quench the torsional motion. When the filter was applied as a set of six pulses with delays of 29, 27, 25, 21 and 19 time-steps (corresponding to the five predominant frequencies) the dihedral angle trajectory and spectral density shown in figure 4.14 resulted. After the quenching filter, the molecule only samples the gauche conformation and new torsional peaks have become prominent. Figure 4.14 also shows the result of a normal mode analysis of the gauche conformation which is in good

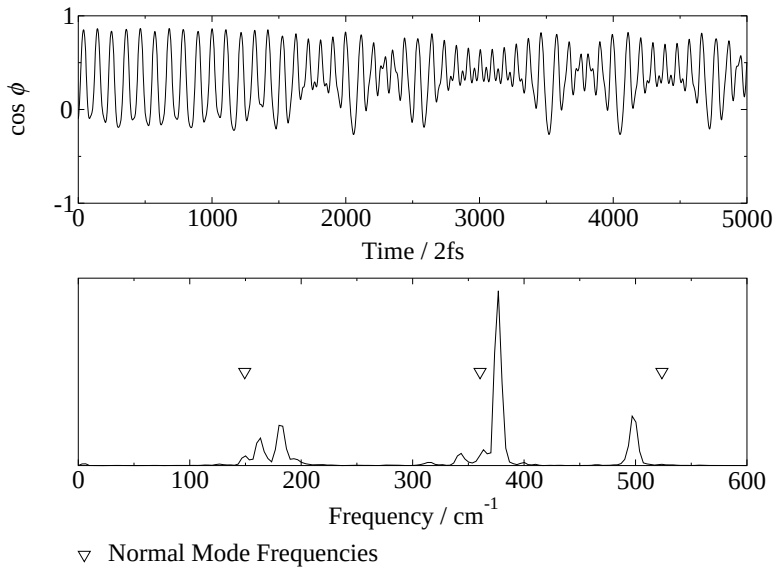

Figure 4.14. Dihedral angle trajectory and spectral density with NM analysis of butane after the amplified torsional motion is partially quenched.

agreement with the spectral density. If it was necessary to stop the torsional motion completely the same filter could be used again with target frequencies of $162\,\mathrm{cm}^{-1}$ and $181\,\mathrm{cm}^{-1}$.

4.3.3: Liquid-Phase Butane

In the condensed phase, the calculation of the spectral density is complicated by the fact that the molecules are translating and rotating. In addition, if the calculation is performed using the atomic coordinate data, then the influence of periodic boundary conditions must be removed. Using the atomic velocity data therefore offers an advantage since there are no discontinuities arising from periodic boundary conditions, and furthermore, the velocities may be readily processed to remove centre of mass translation and rotation. Spectral densities calculated from these processed atomic velocity components will therefore reflect only the frequencies of the internal degrees of freedom. Figure 4.15 shows two spectral density plots of a single molecule obtained from a simulation of liquid butane; one spectrum is calculated from coordinate data and the other from processed velocity data. The two plots are very similar at high frequency but diverge at low frequency where the effect of net molecular translation and rotation becomes apparent.

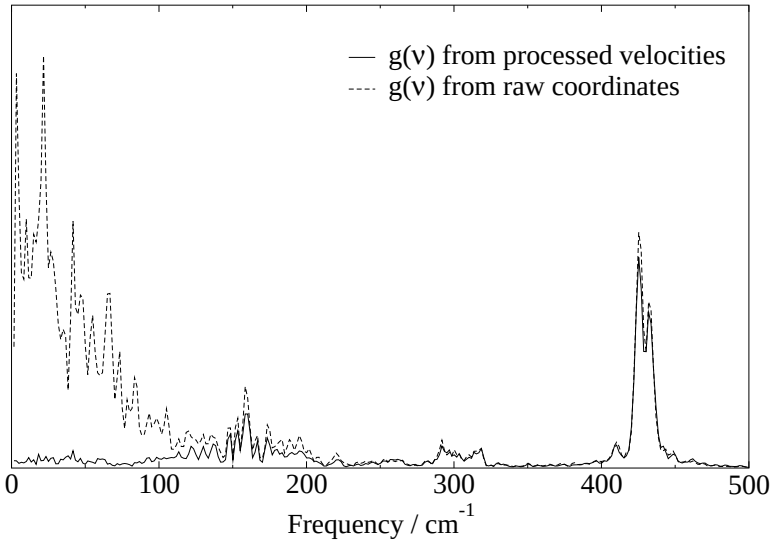

Figure 4.15. The spectral density of one liquid-butane molecule calculated from raw atomic coordinates and processed velocities.

The velocities used in DFMD must be processed in an identical fashion, since it is the velocity components relating to the motion of internal degrees of freedom to which the digital filter must be applied. The centre of mass rotational and translational velocity components are therefore removed from every stored velocity using the following steps:

1. Bring all coordinates for the particular molecule into the same periodic box by translation.
2. Calculate the centre of mass (COM) coordinate and velocity.
3. Translate the COM to the origin and zero the translational velocity.
4. Calculate the angular momentum of each atom about the origin and sum them to give the total angular momentum of the rigid body.
5. Using the inverse of the inertial tensor, calculate the angular velocity of the rigid body.
6. Remove from each atom the velocity due to the angular velocity.
7. Store these modified internal velocities and restore the original velocities and coordinates to continue MD.

Application of the filter will then yield a set of filtered internal velocities, to which the centre of mass rotational and translational velocity components from the filter mid-point should be added. In this way the filter acts on only the internal degrees of freedom. Thus, use of DFMD on the condensed phase requires not only the central coordinate set

to be stored, but also the central centre of mass translational and rotational velocities.

In the liquid phase, the molecules are constantly moving and colliding with each other and their internal motion is therefore complicated by the intermolecular interactions. Thus the trajectory of a single molecule in the liquid phase is more irregular than its gas phase counterpart. To illustrate this point, three spectral densities are shown in figure 4.16, each calculated for a single butane molecule in liquid butane from 4001 processed velocity sets taken from different parts of the same molecular dynamics trajectory. The intensities of the vibrations can vary by an order of magnitude but the frequencies of the motion remain constant. Thus, to apply a filter to just one molecule of the liquid, the time of filter application becomes more important—it would be much easier to achieve a large amplitude motion at $430\,\mathrm{cm}^{-1}$ if the enhancement filter was applied at a point in the trajectory when the amplitude was already large (figure 4.16, solid line). In this more dynamic system the spectral density plots become less useful. The frequency-resolution of a spectral density plot is dependent only on the period of time sampled and therefore to obtain good resolution a long sample is needed. However, in the condensed phase the internal dynamics are rapidly changing, and a short sample time is needed to focus on rapidly changing events. A compromise sample period must therefore be used if the spectral density is to be be converged and yield useful information on the effect of the filter; this may not be possible. For these reasons, internal coordinate plots will be used in what follows to demonstrate the effect of the filter.

To examine the performance of DFMD in the condensed phase, liquid butane simulations were run using the same force field as the second set of gas-phase butane simulations described above. 64 butane molecules were placed in a cubic box of side 21.68 Å. Periodic boundary conditions were applied. The system was equilibrated by running for 50 ps at 270 K in the NVT ensemble. All the DFMD simulations were run in the NVE ensemble with an initial temperature of 270 K and a 2 fs time-step.

Five DFMD experiments were performed: quenching high frequency bend motion, quenching low frequency torsional motion, pumping high frequency bend motion, and pumping low frequency motion with and without driving the torsion into the gauche conformation. All filters consisted of 4001 coefficients and were applied to only 1 of the 64 molecules. Target frequencies were selected from the spectral densities of the

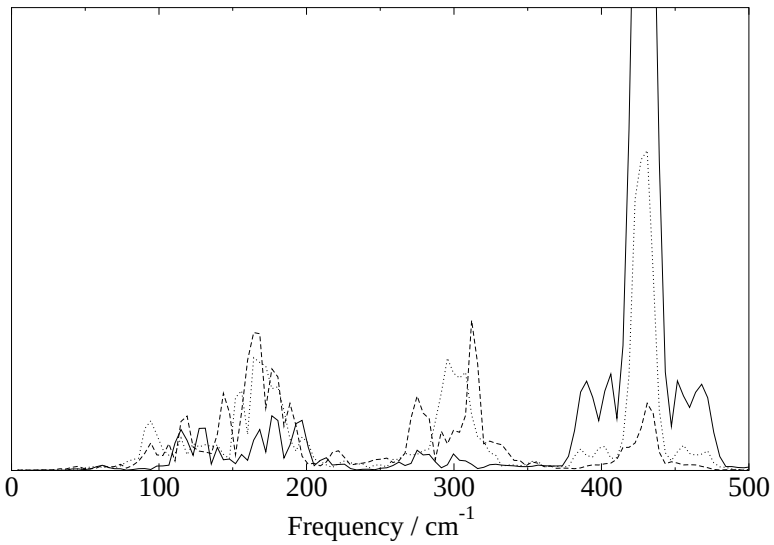

Figure 4.16. The spectral densities of one liquid-butane molecule calculated from three different points in the MD trajectory.

unfiltered molecular dynamics trajectories shown in figure 4.16.

To quench the high frequency bend motion, the peak at $430\,\mathrm{cm}^{-1}$ was targeted with a filter for which $\mathcal{H}(\omega) = 1$ for $\omega < 364\,\mathrm{cm}^{-1}$ and 0 otherwise. Two pulses were used with a pause of 10 time-steps. As a result of the filter's action the bend amplitudes were significantly decreased but some residual motion at lower frequency remained. A more broad-ranging filter for which $\mathcal{H}(\omega) = 1$ for $\omega < 236\,\mathrm{cm}^{-1}$ and 0 otherwise was used to try to remove all high frequency motion. Six pulses were used with delays of 15, 14, 13, 10 and 9 time-steps corresponding to target frequencies of 278–$463\,\mathrm{cm}^{-1}$. The amplitude of the bond angle vibration immediately after the filter application is very small, demonstrating the success of this scheme. In both cases, although the trajectory of the dihedral angle was initially unaffected, it ultimately diverged from the unfiltered trajectory as a consequence of energy redistribution and coupling to intermolecular motion. Figure 4.17 illustrates one of the bond angle trajectories when it is (a) unfiltered, (b) filtered by the double-pulse and (c) filtered by the six-pulse application. It is apparent from these figures that kinetic energy slowly re-enters the angle bending motion as a consequence of coupling to intermolecular motion. This effect was not observed in the gas-phase calculations.

Quenching the low frequency torsional motion proved to be less satisfactory. This is due to a combination of two factors: fast energy redistribution through coupling to low

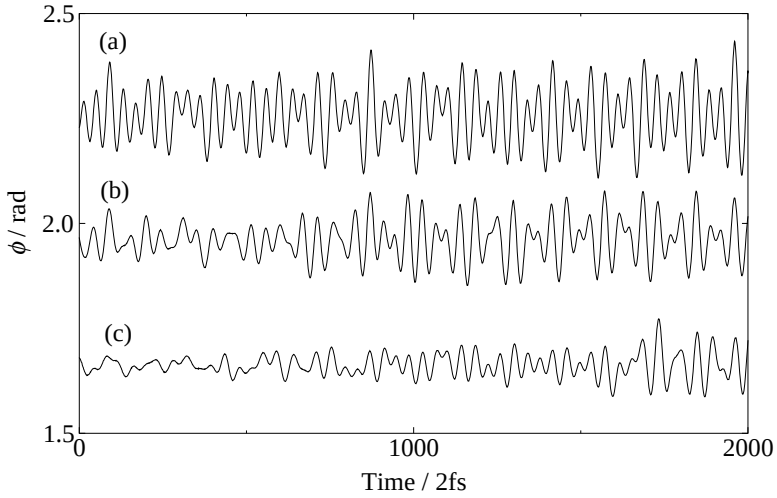

Figure 4.17. Bond angle trajectories: (a) unfiltered (offset by 0.3 rad), (b) double-pulse quench filter, (c) six-pulse quench filter (offset by −0.3 rad).

frequency intermolecular motion, and a broad undefined low-frequency target. Both of these factors arise from the unavoidable nature of the condensed phase and combine to make quenching more difficult.

To pump energy into the bend motion, a double-pulse filter with $\mathcal{H}(\omega) = 3$ for $\omega > 364\,\mathrm{cm}^{-1}$ and $\mathcal{H}(\omega) = 1$ otherwise was used with a 10 time-step delay. The effect of the filter on the dihedral and bond angle trajectory is shown in figure 4.18. In comparison to the unfiltered trajectory, the bond angle vibrational amplitude has more than doubled and the dihedral angle trajectory is unchanged. The bond angle vibrational amplitude starts to decrease back towards the unfiltered value after approximately 2 ps and it is only then that the dihedral angle trajectory diverges from that of the unfiltered simulation.

Adding energy to the dihedral motion proved to be easier than in the gas-phase. The combination of a broad peak and faster energy redistribution means that the frequency does not need to be targeted as precisely. The torsion was first targeted with a double-pulse filter that had $\mathcal{H}(\omega) = 2$ for $\omega < 240\,\mathrm{cm}^{-1}$ and $\mathcal{H}(\omega) = 1$ otherwise. The pulse delay was 26 time-steps, corresponding to a frequency of $160\,\mathrm{cm}^{-1}$. Figure 4.19 illustrates the filter's effect on the dihedral and bond angle motion. In comparison to the unfiltered trajectory the amplitude of the dihedral motion is significantly increased for six cycles before returning to that of the unfiltered trajectory. The amplitude of the

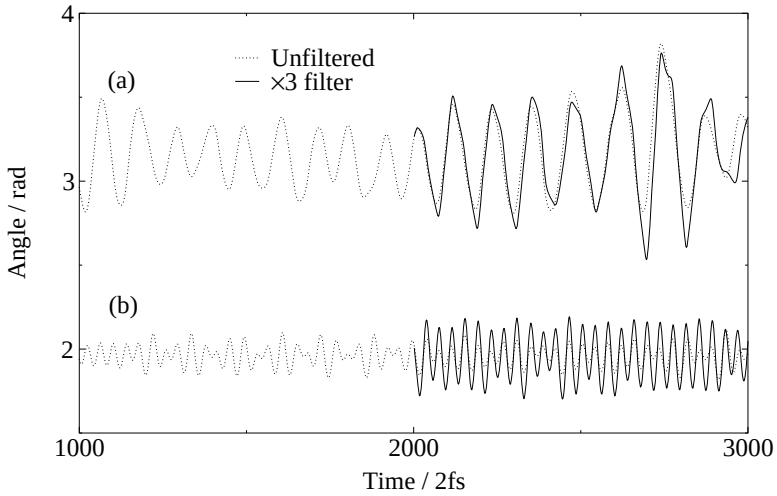

Figure 4.18. The effect of a ×3 bond angle amplification filter on (a) the dihedral angle, and (b) the bond angle trajectory.

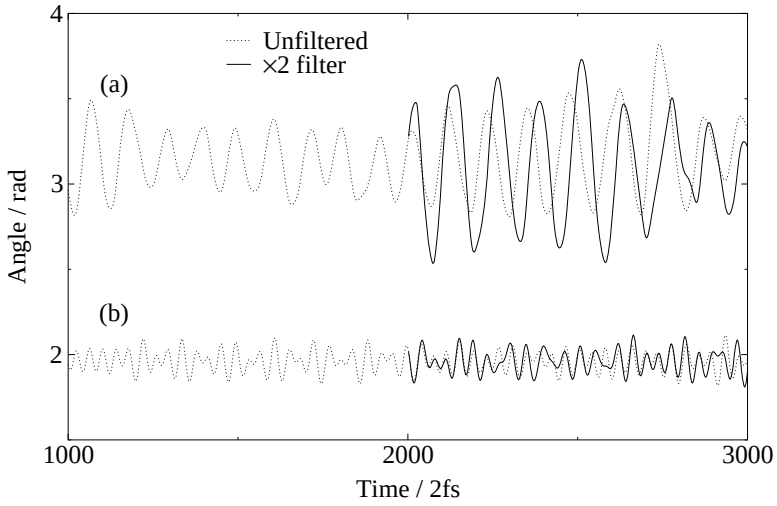

Figure 4.19. The effect of a ×2 torsion amplification filter on (a) the dihedral angle and (b) the bond angle trajectory.

bond angle oscillation is, as expected, unaffected.

In a second simulation the dihedral angle was targeted with a ×6 amplification filter for which $\mathcal{H}(\omega) = 6$ for $\omega < 240\,\mathrm{cm}^{-1}$ and $\mathcal{H}(\omega) = 1$ otherwise. The same pulse delay of 26 time-steps was used and this time the dihedral angle moved cleanly from the trans conformation into the gauche conformation (figure 4.20). In comparison to the gas-phase (figure 4.13) this conformational change was very smooth because of the coupling to the intermolecular degrees of freedom—once the molecule reaches the gauche conformation the dihedral kinetic energy dissipates and the molecule stays in the gauche conformation.

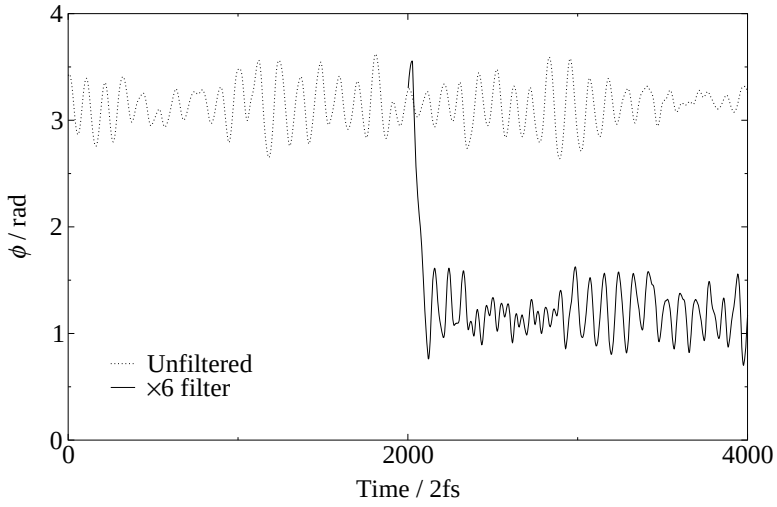

Figure 4.20. Liquid butane dihedral angle trajectory showing the effect of a ×6 torsion amplification filter.

4.3.4: Prion Protein

To examine the performance of DFMD in enhancing conformational change in a protein system, molecular dynamics simulations of the Syrian hamster prion protein were performed. The course of prion disease is believed to involve the conversion of the prion protein from the native conformation that is predominantly α-helical to one containing a significant proportion of β-sheet.[9, 10] Certain amino acid mutations are known to predispose individuals to prion disease, and more specifically, simultaneous changes at positions 129 and 178 determine the type of prion disease involved.[86] Molecular dynamics computer simulations have shown that the distance between these two residues is quite large, but that Tyr 128 is directly hydrogen bonded to Asp 178.[87] This suggests that the role of mutations at positions 129 and 178 may be indirectly influenced or mediated by a residue 128 to 178 interaction. DFMD has therefore been applied to this system to determine whether this potentially important hydrogen bond interaction may be selectively disrupted.

Molecular dynamics computer simulations were performed on Syrian hamster prion protein using the AMBER simulation package[17]. A protein structure equilibrated for approximately 1 ns at 293 K was taken from an earlier molecular dynamics simulation; the results of this simulation have been reported elsewhere.[87] The simulation system contains 4790 TIP3P water molecules[88] and a single Na^+ counter-ion. The all-atom parm96 parameter set[89] was used in conjunction with cubic periodic boundary conditions. All

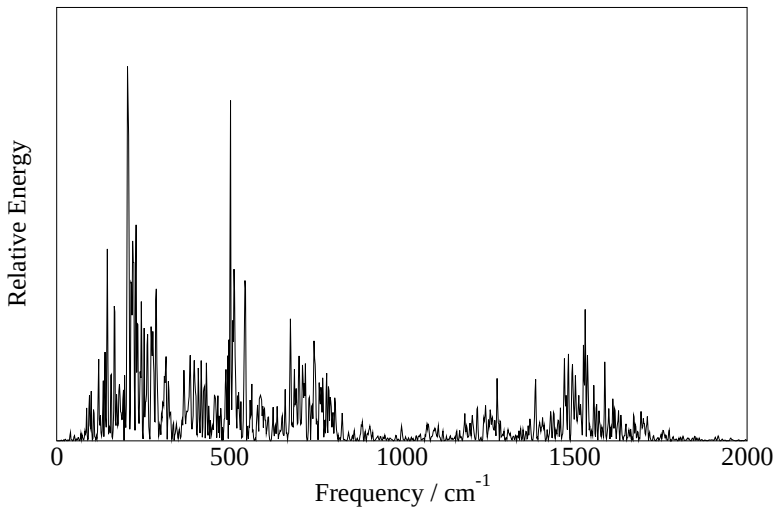

Figure 4.21. The spectral density of the seven prion protein atoms described in the text

bond-lengths were constrained using the SHAKE[8] algorithm and a time-step of 2 fs was employed. Long-range electrostatic interactions were included using the particle mesh Ewald method,[23] and for the Lennard-Jones interactions a 12 Å residue-based cutoff was used. The DFMD simulations were run at constant NVE.

In the simulation there is a hydrogen bond between the hydrogen atom of the hydoxyl group on Tyr 128 and one of the two oxygens of Asp 178. This hydrogen bond was chosen as an interesting target for DFMD. The target atoms for the filter were the Asp 178 oxygens, the hydroxyl group and three nearest ring carbons of Tyr 128. In figure 4.21 the spectral density for the target atoms is presented, accumulated over 8001 time-steps. The spectral density is very complex and does not help very much in deciding upon the target frequency. Figure 4.22 shows the spectral density of the hydrogen bond distance. This spectral density was calculated using equation 3.10 and therefore cannot be compared quantitatively with the spectrum in figure 4.21. The two figures are very similar, demonstrating that almost all the motion in the seven atoms is also present in the hydrogen bond.

Neither of the spectral densities are very useful in determining which frequency to target. Figure 4.23 shows the amplitude spectrum of the hydrogen bond distance. The spectral density of a single degree of freedom may be calculated from the amplitude spectrum by multiplying each amplitude by the square of the frequency. This amplitude spectrum clearly shows that the low frequency motions have by far the largest

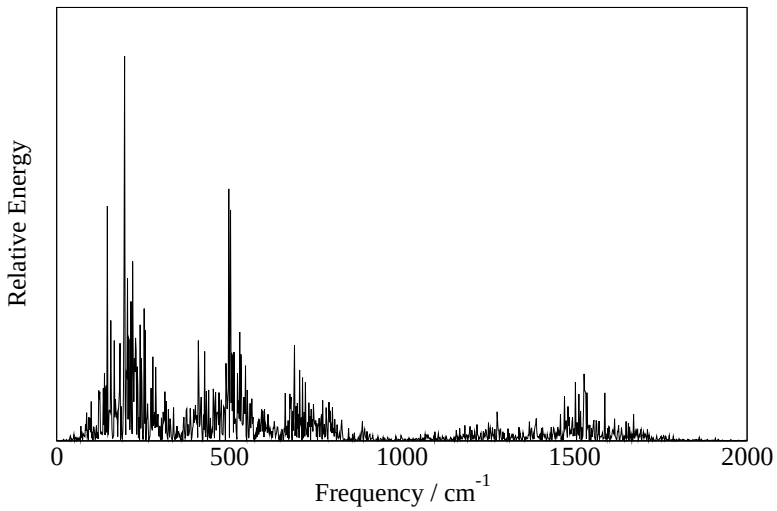

Figure 4.22. The spectral density of the variation with time of the hydrogen bond distance between Tyr 128 and Asp 178.

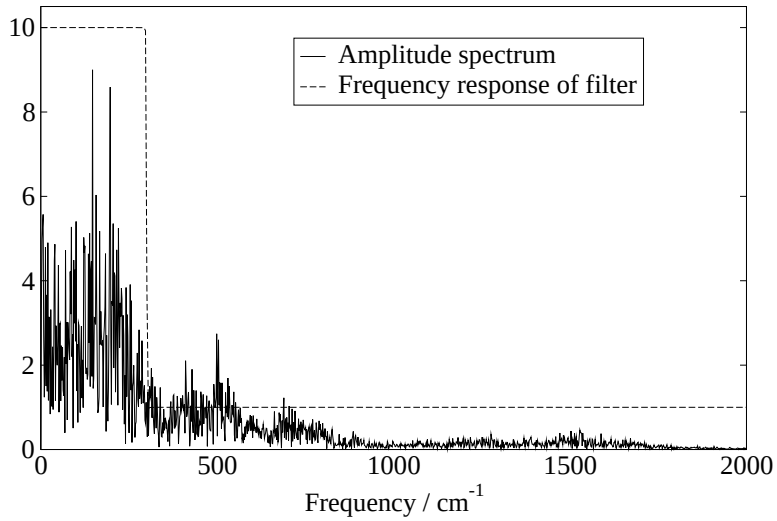

Figure 4.23. The amplitude spectrum of the variation with time of the hydrogen bond distance between Tyr 128 and Asp 178 (arbitrary y-axis scale) and the frequency response of the digital filter applied.

amplitude and it is therefore the low frequency region which was chosen as the target for amplification.

To enhance the motion associated with the low frequency region, a filter for which $\mathcal{H}(\omega) = 10$ for $\omega < 300$ and 1 elsewhere (shown in figure 4.23) was applied using 8001 coefficients. Seven pulses were used with delays of 83, 41, 28, 21, 17 and 14 time-steps, corresponding to target frequencies of 50, 100, 150, 200, 250 and 300 cm^{-1}. In figure 4.24, the distance between the hydrogen atom of Tyr 128 and both oxygen atoms of

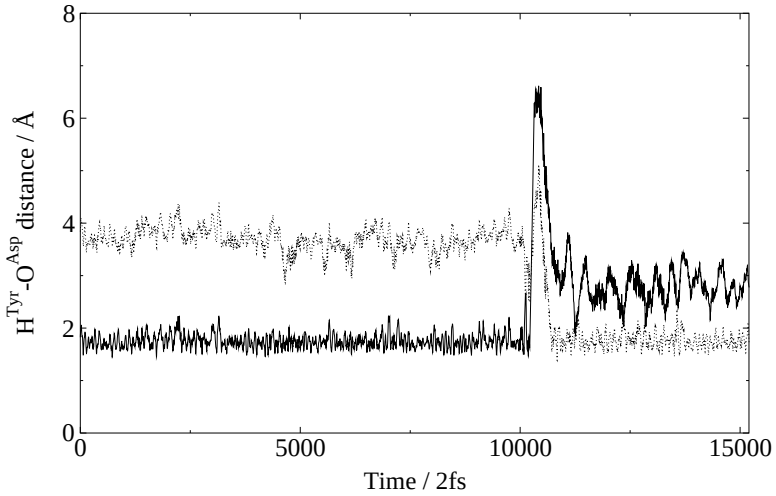

Figure 4.24. The two Tyr 128 - Asp 178 H-bond distances as a function of time. The 0–300 cm^{-1} $\times 10$ filter is applied at step 10002.

Asp 178 is presented; the first digital filter pulse is applied at 10002 time-steps. Before the first pulse, a stable hydrogen bond is observed with one of the oxygen atoms. After the digital filter has been applied, the hydrogen bonding pattern is completed disrupted and a new hydrogen bond forms with the other oxygen atom. In figures 4.25 and 4.26, the structure of this section of the protein is presented before and after the digital filter has been applied. It is apparent that the conformations of the two amino acid side chains have changed considerably and that this critical interaction has been selectively disrupted.

To examine whether non-selective heating of these amino acid residues could have achieved the same response, the velocities of the same subset of atoms were scaled by an empirical factor of 3.5 using the same pulse sequence as above. This has the effect of increasing the total energy of the system to within 4% of the change caused by the digital filter, but this time the kinetic energy is added non-selectively. In figure 4.27, the distance between the hydrogen bonded atoms is shown; the heating pulses are applied starting at 10002 time-steps. It is apparent from this figure that although the addition of thermal energy perturbs the hydrogen bonding, it is insufficient to break the hydrogen bond. These results demonstrate that the selective addition of kinetic energy using a digital filter is clearly more effective than simple heating.

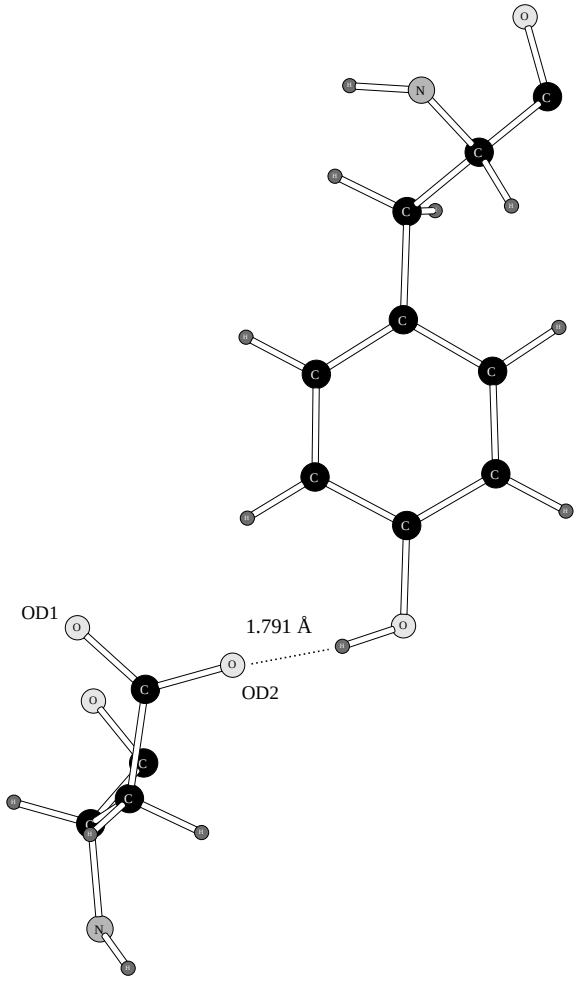

Figure 4.25. Typical conformation of Tyr 128 and Asp 178 before the filter application.

4.4: Conclusion

The new technique of Digitally Filtered Molecular Dynamics has been developed. DFMD involves the application of digital filters to modify the velocities in a molecular dynamics simulation. The modification is such that vibrations within the system are either enhanced or suppressed in a controlled fashion according to frequency. The application of this methodology to four test cases, water in the gas phase, butane in both gas and liquid phases and the Syrian hamster prion protein, has been described, and in particular it has been shown that DFMD can bring about conformational change selectively. In the case of the prion protein, it has also been shown that DFMD is more efficient than the simpler alternative of non-selective heating. The simple examples presented in this chapter clearly demonstrate the remarkable selectivity and control

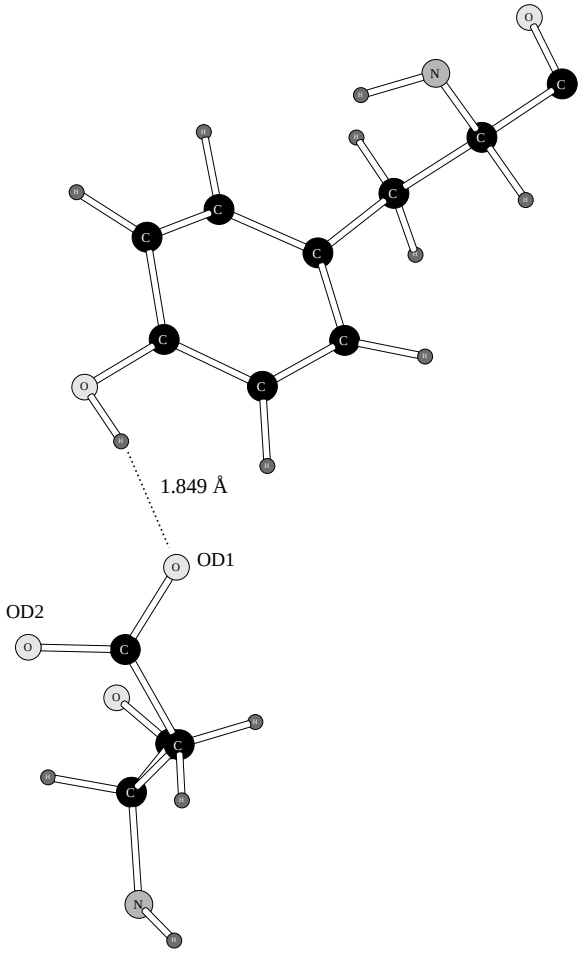

Figure 4.26. Typical conformation of Tyr 128 and Asp 178 after the filter application.

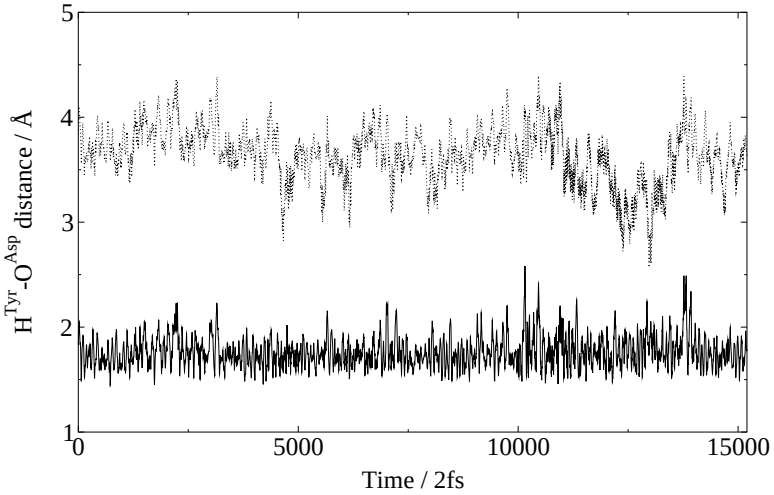

Figure 4.27. The two Tyr 128–Asp 178 H-bond distances as a function of time. The non-selective heating pulse sequence starts at step 10002.

possible using DFMD.

The work presented in this chapter has been published in the Journal of Chemical Physics.[90]

5: Reversible Digitally Filtered Molecular Dynamics

The DFMD method may be used to add or remove energy from chosen frequencies using a timed double-pulse filter application. While DFMD is a demonstrably useful method, it does have a number of limitations:

1. Trying to target a low frequency in a system where the energy dissipates rapidly cannot be achieved using the timed double filter pulse technique. This is because, for a low frequency, the pause between the two pulses is so large (see equation 4.4) that the energy introduced by the first pulse has dissipated before the second pulse is applied.

2. All the energy required to produce conformational change must be introduced with a single double (or multiple) pulse application. Any subsequent filter application cannot be made until the filter buffer is refilled, during which time any effect of the first filter application will have dissipated with the result that the second application starts effectively from scratch. The energy dissipation half-life in solvated alanine dipeptide and in the pentapeptide YPGDV (presented in this chapter) was found to be of the order of 15 time-steps. It is therefore not possible to put energy into the

system gradually—the filter has to be applied to a trajectory that is only sampling the bottom of a potential well. The characteristic motion of the bottom of a potential well cannot be expected to be similar to the motion required to move through a transition state.

In this chapter, the Reversible Digitally Filtered Molecular Dynamics technique (RDFMD) is presented. RDFMD overcomes these problems by providing complete freedom over the timing of the filter pulses. With RDFMD, frequency-targeted energy may be introduced into the system gradually, providing the following benefits:

1. If the energy is added gradually then the trajectory will be able to sample the potential surface more fully, rather than going directly from sampling the bottom of the well to going straight over a transition state as may happen with DFMD. By first sampling higher up a potential well, the eventual transitions may be more reasonable.
2. The characteristic frequency of an oscillation may change as the system samples higher up the potential surface. By putting the energy in gradually, the target frequency may be adjusted to match the transition frequency.
3. Putting the energy in gradually may make it easier to get close to the minimum amount of extra kinetic energy required for a transition.

5.1: Theory

How can we put the energy in more gradually? Before answering this question it will be instructive to recap on how DFMD works. Figure 5.1 shows the sequence of simulations and filter applications for a DFMD simulation applying the filter as often as possible (the phrase "applying the filter" encompasses single or multiple-pulse applications). Normal MD is performed followed by a period of MD during which the filter buffer is filled. The filter is then applied, taking the simulation back to the mid-point of the buffer. To apply the filter again as soon as possible we start filling the buffer immediately.

Reversible Digitally Filtered Molecular Dynamics (RDFMD) is an extension to the basic DFMD method which lets the second and subsequent filter applications be positioned anywhere in time relative to the first application. Thus it is possible to apply a small amplification filter several times while moving the application point forward in

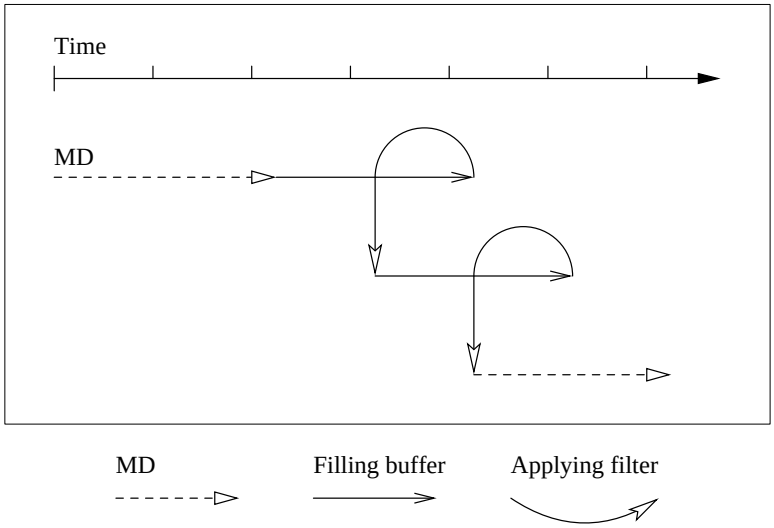

Figure 5.1. Applying the DFMD method twice, with no pause in between.

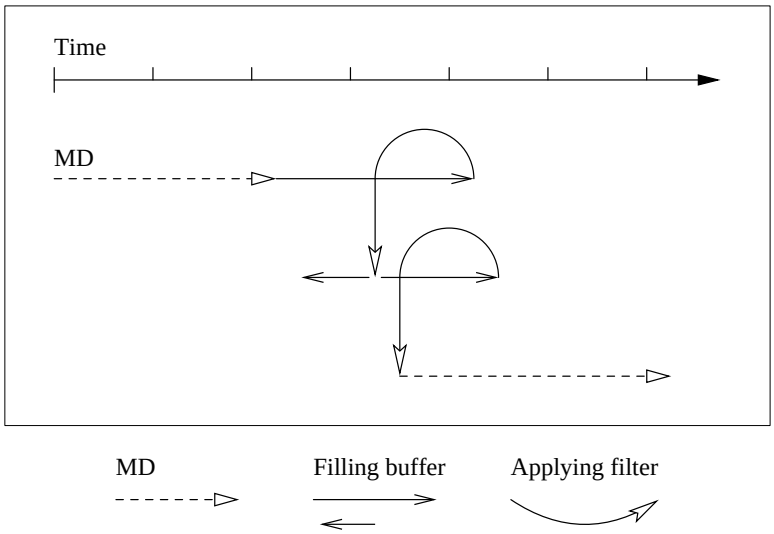

Figure 5.2. The RDFMD method. When filling the buffer the MD proceeds both forwards and backwards in time, allowing complete control over the positioning of subsequent filter applications.

time only slightly or not at all. The scheme is illustrated in figure 5.2.

When a filter is applied, the coordinates and forces are restored from the centre of the filter buffer. In RDFMD, the position of the centre of the filter buffer for the second and subsequent applications is specified by the user. This is achieved by filling the new buffer from MD simulations run both forwards and backwards in time after the first filter application. Running the simulation backwards in time is achieved by simply changing the sign of the time-step. Figure 5.2 shows a typical example of the

RDFMD method. Normal MD is used first, followed by a period during which the filter buffer is filled. The filter is then applied, taking the coordinate set from the mid-point of the filter buffer. Up to this point, RDFMD is exactly the same as DFMD. Next, two simulations are run, both continuing from the initial filter application: one simulation proceeds forwards in time while the second simulation changes the sign of the time-step and goes backwards in time. The filter buffer is refilled, with the first part of the buffer taken from the backwards simulation and the rest of the buffer taken from the forwards simulation. The filter is applied in the usual way with the system state being restored from the centre of the buffer. Normal MD or further filter applications may follow.

By varying the proportion of the new buffer taken from the backwards simulation one can choose the position of the new centre-point. For instance, if the filter buffer consisted of 8001 coefficients and simulations were performed both backwards and forwards for 4000 steps, then the centre point of the new buffer would be in the same place as before. However, by simulating backwards for 3990 steps and forwards for 4010, the centre point of the new buffer is moved 10 steps forwards. The effect is to yield a full filter buffer from an amplified trajectory but with a choice of filter buffer centre point. By choosing the centre point of the new filter buffer to be near the centre point of the old buffer, the filter application is able to build on the energy added to the system by previous applications, and energy dissipation effects can be minimised.

To simulate backwards in time, a time-reversible integrator is required. The velocity Verlet[91] is such an integrator and is implemented in both AMBER version 6 (ref. 17) and DLPROTEIN (ref. 19). The form of the velocity Verlet integrator used in DLPROTEIN is taken from Martyna *et al.*[92] The standard form of the integrator (which is used by DLPROTEIN in the NVE ensemble) is given here:

$$\begin{aligned} \mathbf{r}(t+\delta t) &= \mathbf{r}(t) + \delta t \mathbf{v}(t) + \frac{1}{2}\delta t^2 \mathbf{a}(t) \\ \mathbf{v}(t+\delta t) &= \mathbf{v}(t) + \frac{1}{2}\delta t[\mathbf{a}(t) + \mathbf{a}(t+\delta t)] \end{aligned} \tag{5.1}$$

The previous simulations have used the leap-frog Verlet integrator (see section 2.1). The velocity Verlet calculates and stores the positions and velocities at each time step (rather than holding the velocities half a time-step out of phase, as in the leap-frog

Verlet, or not at all, as in the original Verlet). This means that it is trivial to reverse the integrator by changing the sign of either the time-step or the velocities.

Is it valid though, to reverse the integrator and collect data from the reversed simulation? When running in the NVE ensemble with the velocity Verlet integrator, if the sign of the time-step is changed (and the simulation restarted) then the simulation exactly retraces the reverse of the forward trajectory (to the limit of the machine's numerical precision). In RDFMD we use the NVE ensemble and simulate for a small amount of time in reverse. Only velocity data from the backwards trajectory is used for the purpose of extending the filter buffer backwards in time.

DLPROTEIN (ref. 19) was chosen as the program for implementing RDFMD. DLPROTEIN is a modification of the DL_POLY (ref. 18) code which is intended to be used specifically for modeling protein systems. It inherits the clear style and documentation from DL_POLY (although the documentation for the alterations is somewhat deficient). To implement the new scheme required a significant rewrite of a lot of the DFMD code. To apply the filter for the first time, the buffer is filled, the filter is applied and a restart file is saved. To apply the filter again, a reverse simulation begins from the restart file but with a negative time-step. The velocity information is stored in a file. The forward part of the simulation then begins from the same restart file, firstly reading in the velocity data from the reverse simulation (placing it in the correct place in the filter buffer), and then moving forward in time until the buffer is full. The filter is then applied and another restart file is saved. The filter application itself can either be a single pulse or a sequence of multiple pulses as before. Several scripts have been written to automate these simulations.

5.1.1: Filter Design

Section 3.3.2 discussed the Fourier filter design method and sections 3.3.3 and 3.3.4 explored the properties of filters including the windowing of a Fourier-designed filter. The Fourier design method in combination with the von Hann window is a basic design technique which served its purpose well. At this stage of the work a second study of filter design was undertaken to see if any improvement could be made. Specifically, it was asked whether a similar quality filter could be designed using fewer coefficients and

thus reduce the overhead associated with the application of the filter.

The MATLAB (ref. 93) application provides a Signal Processing Toolbox that contains many filter design algorithms. MATLAB places the various FIR filter design methods into three categories: "Windowing", "Multiband with Transition Bands" and "Constrained Least Squares". The windowing category comprises methods that apply a windowing function to the Fourier-designed filter, as discussed already. "Multiband with transition bands" design methods provide a more general means of specifying the ideal desired filter than the windowing methods. They allow transition regions in which frequency response is not considered important to be specified. The "Constrained Least Squares" (CLS) filter design functions enable you to define upper and lower thresholds that contain the maximum allowable ripple in each region of the filter. This provides the greatest flexibility of the three classes of design method.

To compare the various design methods, a desired frequency response of $\mathcal{H}(f) = 10$ for $f < 0.5$ and 1 otherwise was chosen. This is similar to the filter used in the prion protein work. The frequency range of f in MATLAB is normalised to be 0–1. Figure 5.3 compares the frequency responses of five filters, all designed using 201 coefficients. The first filter on the list is designed using the *fir2* function—one of the windowing methods—and the parameters have been chosen to exactly mimic the method previously used in this work (i.e. the Fourier design method with a von Hann window). The remaining filters all use the *firls* method from the "multiband with transition bands" category. The design commands for the filters are shown in the figure. The first parameter in the *fir2* command is one less than the number of coefficients required. The second parameter is a vector, **f**, of normalised frequencies. The third parameter is a vector of the desired magnitude response at each frequency in **f**. The final (optional) parameter is the window to be applied to the design. The *fir2* filter shown in 5.3 uses the command `fir2(200, [0 0.49999 0.50001 1],[10 10 1 1], hanning(201))`. This means that the filter will be 201 coefficients in length and will ideally move abruptly from an response of 10 in the region 0–0.49999 to 1 in the region 0.50001–1. A Hanning (or von Hann) window of length 201 will be applied to the design. The parameters for the *firls* command are the same as the *fir2* command except that, as *firls* does not use a window function, no parameter for the window is used. The first *firls* filter has a transition region almost

the same shape as the *fir2* filter, demonstrating that the method can do at least as well. Going down the list, the width of the transition region between the $\times 10$ and $\times 1$ regions is reduced. However, in constraining the transition region more, the ripples in the other regions increase. Ripples in the $\times 1$ region, which is magnified in the bottom part of figure 5.3, are particularly undesirable. If a region is targeted for amplification then it is not too crucial whether it is amplified by 9 or 10 or 11, but in the regions that are not being targeted, the frequency response should be as close to unity as possible so that the amplitudes are not affected. The *firls* function can produce a narrower transition region for the same number of coefficients as *fir2* but at the expense of undesirable ripples. The *fir2* function could also produce equivalent filters if a different window was applied.

The fundamental problem is that a given number of coefficients can only constrain the frequency response so much. The *fircls* function is a CLS method that allows the user to specify which regions of the filter are important to have constrained to a certain value and which regions of the filter can be relaxed to compensate. The first parameter of the *fircls* command is one less than the number of coefficients required. The second parameter is a vector of normalised frequencies representing the edges of the regions in the filter (so a vector of order three specifies two regions). The third parameter specifies the desired magnitude response in each region. The fourth and fifth parameters are vectors representing the desired upper and lower bounds on the magnitude response for each region.

Figure 5.4 shows three filters each using 201 coefficients. For reference, the black filter response is the same *fir2* filter as in figure 5.3. The first *fircls* filter listed (shown in red) is defined to be ideally 10 in the region 0–0.5 and 1 in the region 0.5–1. It is permitted to ripple within the ranges 8–10 and 0.983–1.013 respectively (the range for the $\times 1$ region was derived from measuring the variation in the *fir2* filter). Even with the tight constraint in the $\times 1$ region, the large ripples in the $\times 10$ region allow the transition from one to the other to be narrower than the *fir2* filter. The second *fircls* filter constrains the response to be between 10 and 12 for the amplifying region. This has no effect on the width of the transition region.

The fact that the *fircls* function can do better (by our criteria) than the *fir2* function for the same number of coefficients suggests that it can do as well as the *fir2* function

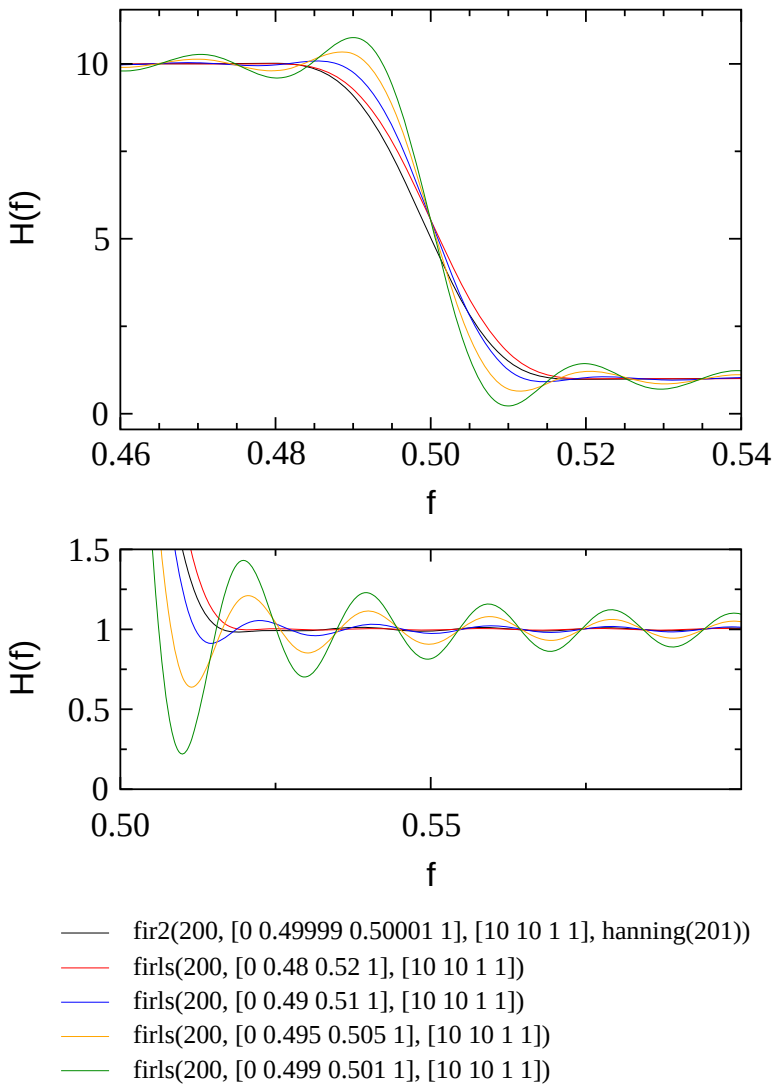

Figure 5.3. Comparing the *fir2* and *firls* functions of MATLAB. The lower plot is a magnified view of the bottom region of the filter. For the same number of coefficients, *firls* can do no better than *fir2*.

with fewer coefficients. Figure 5.5 shows this to be the case. The number of coefficients required to reproduce the transition of the 201 coefficient *fir2* filter was judged by eye and found to be 141. Subsequently, the width of the 9.55–1.45 region (the middle 90% of the transition) for both filters was measured and found to be the same.

To summarise, the *fircls* function provides the means to constrain one region while relaxing another. For an amplifying region the user can then specify that it must amplify by "at least 10" for instance while keeping the $\times 1$ region at an acceptable level. In comparison with the Fourier design and von Hann window used previously, by relaxing

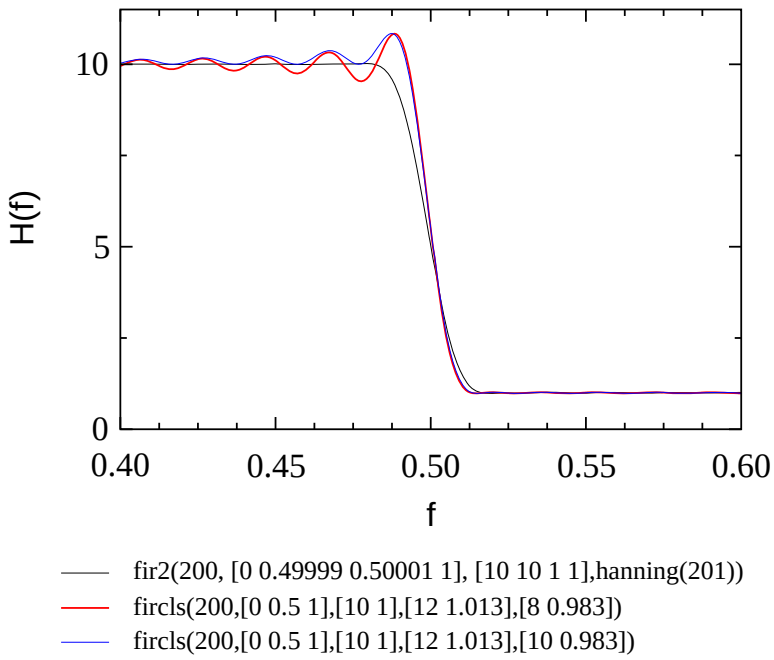

Figure 5.4. Comparing the *fir2* and *fircls* functions of MATLAB. Using the same number of coefficients, *fircls* can produce a narrower transition region without causing undesirable ripples elsewhere.

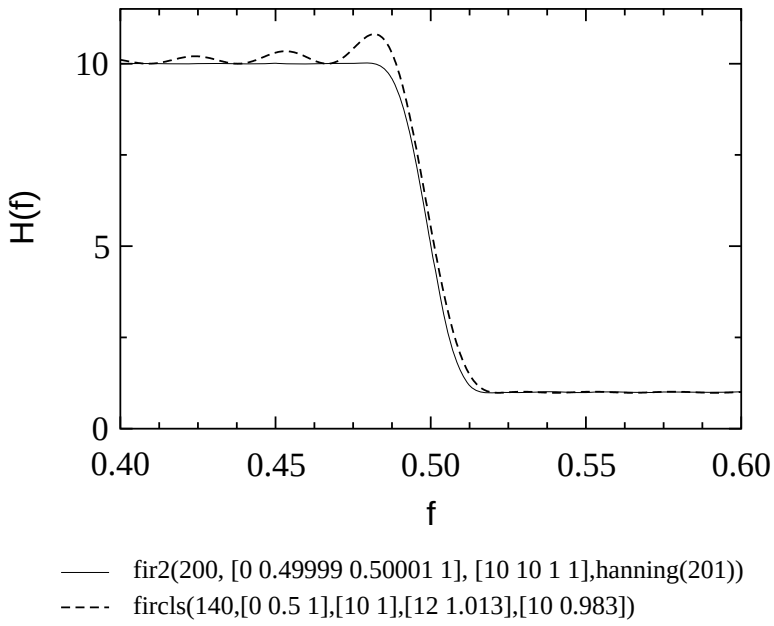

Figure 5.5. Comparing the *fir2* and *fircls* functions of MATLAB. The *fircls* function can produce a filter as good as *fir2* using approximately 70% of the coefficients.

the amplifying region the *fircls* function can produce an equivalent quality filter using approximately 70% of the coefficients.

5.2: Initial Test Cases

While developing and testing the code a simple simulation of a single united-atom butane molecule was used. Once the code worked properly a liquid butane simulation was investigated. The simulations of liquid butane by RDFMD were successful. Energy could be put into the system gradually until conformational change was caused. Unfortunately, as the potential surface for the single dihedral is so simple, and the energy dissipation in the simulations so slight, it was not possible to show any advantage of RDFMD over the basic DFMD method.

5.3: Pentane

A single united-atom pentane molecule was the next target. This was chosen because it was still simple to analyse but had a more complicated potential-energy surface. The parameters for the force field were derived from those in the 1984 OPLS paper by Jorgensen *et al.*[85] This paper derives dihedral potential functions for several liquid hydrocarbons by fitting to MM2 molecular mechanics calculations. For *n*-pentane, a Fourier series is used for each dihedral along with intra-molecular Lennard-Jones potentials for the 1–4 and 1–5 interactions that are different to the intermolecular Lennard-Jones potential parameters. For the sake of simplicity, the special 1–4 and 1–5 interactions were discarded and the Fourier series taken to be the true potential. The Fourier series needed re-parameterizing to yield the same energy curve when the scaled intra-molecular Lennard-Jones potential was included. The re-parameterization was done using BOSS to drive around the united atom CH_3–CH_2–CH_2–CH_2 torsion, once with the original Fourier potential, and once with just the Lennard-Jones potential. The curve from the second dihedral drive was subtracted from the first and a new Fourier series fitted to the difference. In deriving the potential this way we are also assuming that it is transferable to the full 5-atom *n*-pentane.

To create a potential energy surface for the dihedrals, BOSS was used to drive both dihedrals from 0–360° in 5° steps and minimise the other degrees of freedom at each point using the Fletcher-Powell minimiser. Perl scripts were written to plot the potential energy surface and to plot trajectories on the surface. Intermediate points on the surface are estimated using bi-linear interpolation. The potential energy surface is shown in

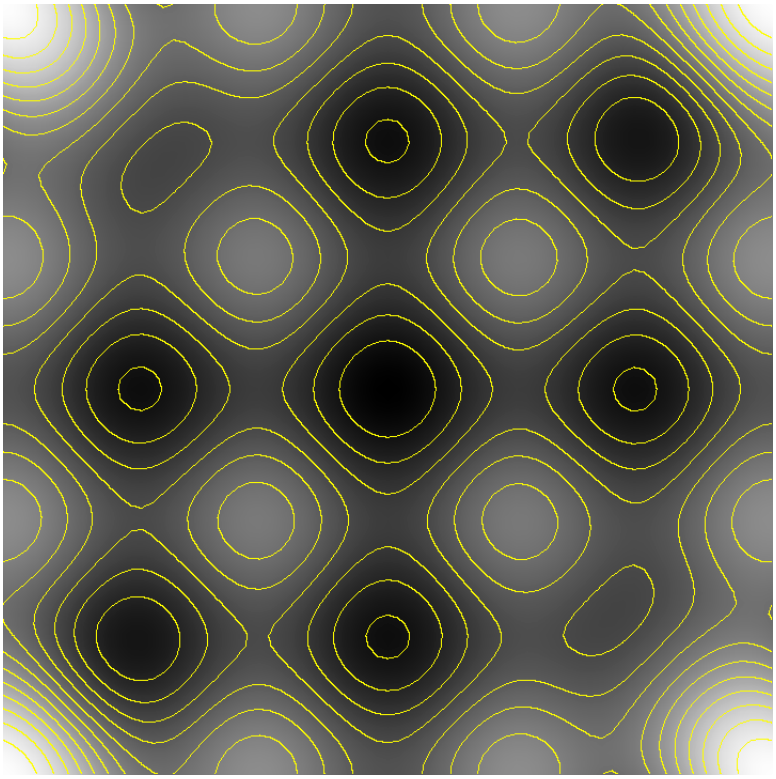

Figure 5.6. Adiabatic potential energy surface for pentane as the two dihedral angles are varied through 0–360°. Dark parts are low energy, bright parts are high energy. Contours are plotted every 1 kcal mol^{-1}.

figure 5.6. The lowest energy conformation (trans-trans) is located in the centre of the figure.

The molecular dynamics simulation was started with the molecule in its minimum energy conformation in the NVE ensemble at an initial temperature of 300 K. All bonds were constrained using SHAKE (ref. 8) and a 2 fs time-step was used. An 8000 step simulation was run and the velocity data was analysed to give a spectral density with approximately 2 cm^{-1} resolution. The spectral densities of the two dihedral trajectories were calculated and summed. In this case it is permitted to take the sum of the spectral densities of the internal degrees of freedom, because they must have identical force constants (even though the force constant is unknown). Both spectra are shown in figure 5.7. Figure 5.7 shows very clearly that the only significant energy in the dihedral oscillation is at 134 cm^{-1} and 173 cm^{-1}. There is a tiny peak at 69 cm^{-1} which is too small to be of interest. Various filters were created using 2001 coefficients to amplify the 0–150 cm^{-1} region by different amounts. Just the 134 cm^{-1} peak was targeted because

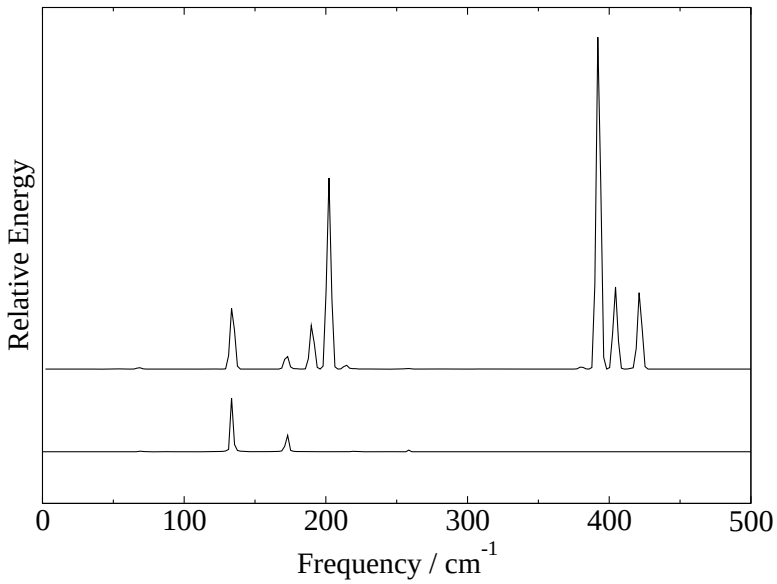

Figure 5.7. Pentane: the spectral density of the internal velocity data (top) and the spectral density of the dihedral trajectories (bottom). The lower graph has been translated in the y-axis in order to make it visible, it has also been scaled in the y-axis for easy comparison with the top graph.

the $173\,\mathrm{cm}^{-1}$ peak is small in comparison and very close to peaks that should remain unaffected.

Many experiments were performed using different filters in different series and with various time delays between applications. The most interesting result arising from the study was a clear difference between putting energy in gradually, and putting it in in a single application. To compare these two methods properly, the simulation series where energy was put in gradually used a zero-step delay between filter applications. The filter was 2001 coefficients in size, so after the initial application, MD was run for 1000 steps backwards and 1000 steps forwards and the centre step of the new filter buffer was the output of the old. By using a zero step delay, we ensure that any difference in the two trajectories is caused by the manner of energy input rather than the time of application. Figure 5.8 shows the dihedral angle trajectory resulting from a single application of a $\times 2$ filter. The figure shows the initial preference of the dihedral angles for moving towards a potential energy maximum rather than a transition point. This preference arises from the initial random atomic velocity assignments.

The simulation series in which the energy was added gradually used nine applications of a $\times 1.2$ filter. The resulting trajectory is plotted in figure 5.9(a). The total

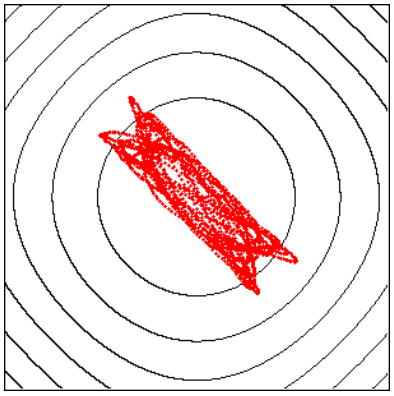

Figure 5.8. Pentane: dihedral angle trajectory resulting from a single application of a $\times 2$ filter. The trajectory is plotted on the 150–210° section of the potential energy surface with contours every 0.5 kcal mol^{-1}.

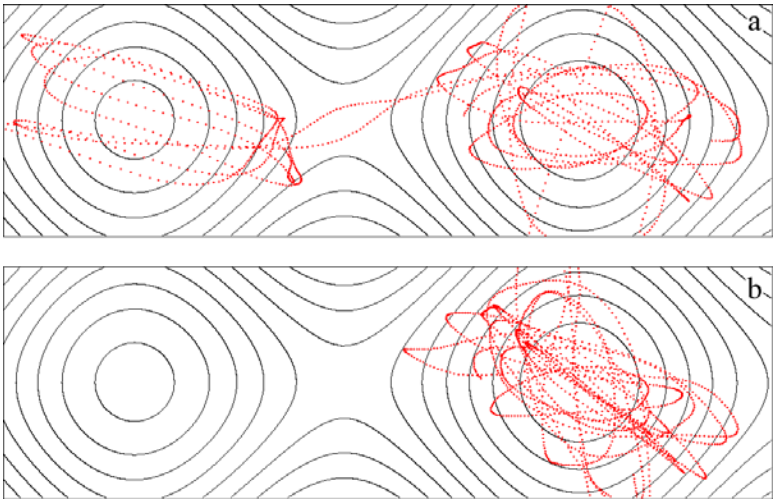

Figure 5.9. Pentane: dihedral angle trajectories plotted on the 30–230° by 150–210° section of the potential energy surface with contours every 0.5 kcal mol^{-1}. Part (a) is the result of gradual energy input. Part (b) results from a single filter application but has the same total energy as (a).

energy after the nine filter applications was 8.92 kJ mol^{-1}. Figure 5.9(b) shows the trajectory resulting from a single application of a $\times 4.6$ filter which gives a total energy of 8.93 kJ mol^{-1}. Adding the energy using a single filter application has not caused a transition. It is clear from the figure that the resulting trajectory spends much of its time moving in a similar manner to figure 5.8, towards the potential energy maximum. In the gradual energy input simulation, the MD has had a chance to explore the higher energy regions of the potential well, causing the final trajectory to move more quickly into a mode suitable for transitions.

In this system, the only way energy can be dissipated from the dihedral degrees of freedom is into the valence angles. This dissipation occurs slowly and so it is likely that the same effect could have been achieved using the basic DFMD technique. The simulations do however demonstrate the benefit of slowly introducing the energy.

Simulations of liquid pentane were then performed, but it proved too easy to cause conformational change in the liquid phase due to the low energy barriers and intermolecular collisions. No further work on liquid pentane was therefore performed.

5.4: Alanine Dipeptide

The next system for study, alanine dipeptide (or *N*-Acetyl-L-alanine-*N*-Methylamide), was chosen because it is more complex, with 12 particles and two distinct and variable dihedral angles, and yet sufficiently simple for a detailed and complete analysis to be performed. It is also a peptide and therefore has the conformational properties of proteins, the ultimate target of this work. Figure 5.10 shows united-atom alanine dipeptide. There have been many simulation studies of alanine dipeptide (often for the same reasons); four already mentioned in the background section (chapter 2) are self penalty walk method,[33] the Olender and Elber paper[49] on long time-step methods (section 2.2.1), the Wu and Wang paper[57] on self-guided molecular dynamics (section 2.2.3) and the leap-dynamics method of Kleinjung *et al.*[67]. Others have studied alanine dipeptide in solution and used various methods for calculating potentials of mean force for its conformations,[94–97] the free energy surface[98] and a conformational probability map.[99] Other studies of conformational transition paths have also been performed.[100, 101]

5.4.1: Potential Energy Surface

The major conformational transitions in alanine dipeptide are changes in the ϕ and ψ backbone dihedral angles (see figure 5.10). In common with other peptides, the ω backbone dihedral angles remain fairly static. A picture of the potential energy surface is necessary in order to understand any conformational change. The surface was calculated using Pepz, BOSS, and various Perl scripts. The united-atom OPLS force field,[1] provided in BOSS, was employed.

Firstly, a Z-matrix was generated by Pepz, part of the MCPRO distribution[7]. This

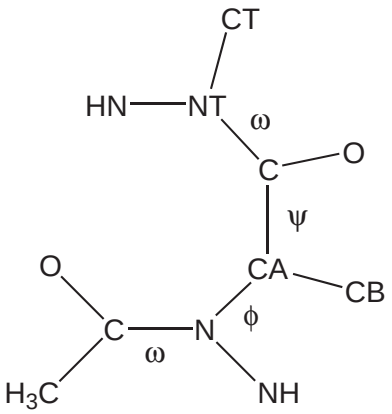

Figure 5.10. United-atom alanine dipeptide showing the ϕ, ψ and ω backbone dihedral angles.

Z-matrix was well suited to a MC simulation where just the ϕ and ψ angles would be varied. However, because all degrees of freedom needed to be minimised, the z-matrix needed modifying to include improper torsions on the backbone atoms. The parameters for these improper torsions were taken from the `parm96.dat` file of AMBER and adjusted to take into account the difference in the force field equation between the two programs.

Even with the correct force field parameters in place, calculating the potential energy surface was problematic. One difficulty resulted from the way the dihedral driving routine in BOSS works. It is possible to instruct the program to vary a dihedral angle systematically from 0–360° in steps of 2° for instance, minimising the structure at each step. However, calculating sections of the potential energy surface in this way produced energy landscapes with abrupt cliffs, where the energy suddenly switched to a very different value in the space of one step. Figure 5.11 shows an example of this where the first 80° are particularly bad. The reason that BOSS fails to find the lowest energy is that the previous minimised structure is used as the starting point for the next minimisation. For example, the structure is minimised with ϕ equal to 180° and ψ equal to -180°. This minimised structure is then adjusted such that ϕ is 178° and the minimisation is run again from this structure. Although this does mean that the minimisations are quick to run, it can cause side-chains to adopt strained conformations which eventually snap back into other minima.

The second problem in calculating the potential energy surface was the choice of minimiser. It was found that using simulated annealing from a starting temperature of

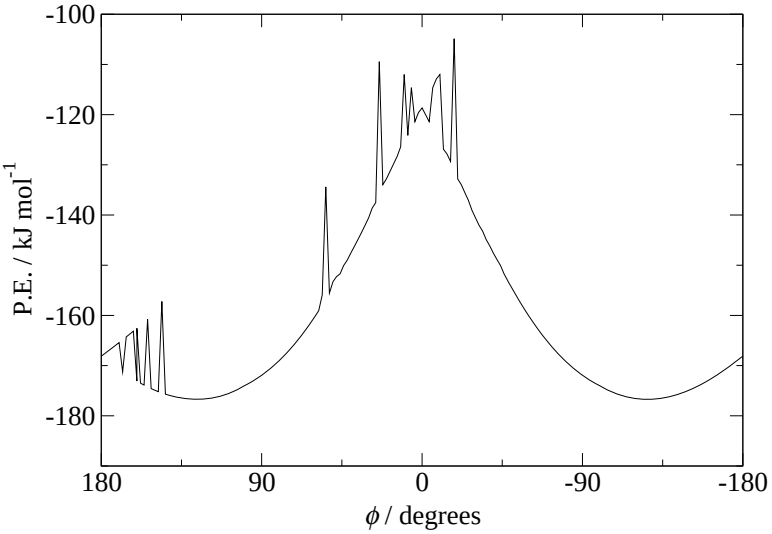

Figure 5.11. A section of the ϕ-ψ potential energy surface for alanine dipeptide. Here ψ is fixed at -180° and ϕ is systematically varied by BOSS from 180° to -180°.

just 40 K caused the dipeptide to switch from one enantiomer to the other, resulting in a rotationally symmetric potential energy surface. To avoid the simulated annealing problem, the simplex minimiser was used. The final surface was calculated by taking the minimum energy at each point of two surface calculations. Both calculations made use of the same unminimised starting structure. In the first calculation a Perl script adjusted the ϕ angle of the starting structure in 2° increments from -180° to 180° and, at each step, called BOSS to drive the ψ angle similarly as described above. This calculation suffers from the snap effects described but was lowest in energy 42% of the time. The second calculation used a Perl script to adjust both the ϕ and ψ angle of the starting structure, covering the whole surface in 2° steps, and called BOSS at each point. This second calculation is slower and does not always find the minimum, but does not suffer from the snap effects of the first. The second calculation was lowest in energy 52% of the time, and the two calculations agreed on the energy for the remaining 6% of the surface.

The final surface is plotted in figure 5.12. The figure was created in Gsharp, part of the UNIRAS package from Advanced Visual Systems. Figure 5.13 shows the same surface but with labels for various minima and transition points. The surface plots and the trajectory plots that follow were produced using the same Perl scripts used for pentane. The surface compares well with the surface published by Elber[49] which was

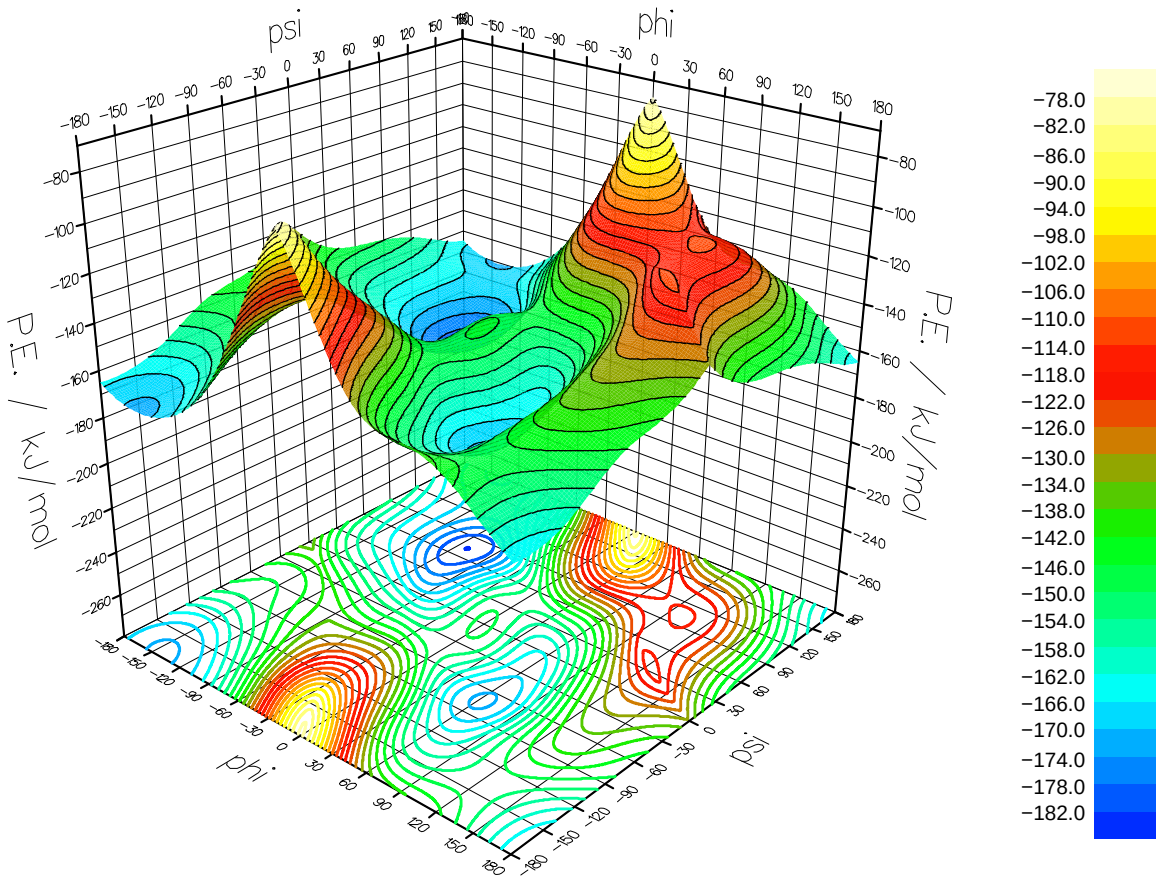

Figure 5.12. Adiabatic potential energy surface of alanine dipeptide. The ϕ and ψ angles are fixed and the rest of the degrees of freedom are minimised.

also calculated using the AMBER/OPLS united-atom force field.

5.4.2: Inducing Conformational Change

For the molecular dynamics simulations, alanine dipeptide was initially placed in minimum B (see figure 5.13). Minimum B is not the lowest energy minimum but it is the deepest, in that the potential energy barrier to reach another minimum is highest. The force field parameters used in the surface calculation were manually translated into the DLPROTEIN format and various dummy dihedral potentials were added to coerce DLPROTEIN into using the same force field as BOSS. All bonds were constrained by SHAKE (ref. 8) and a 2 fs time-step was employed. The initial simulation of the single alanine dipeptide molecule was run for 20 000 steps (or 40 ps). The ensemble was the NVE micro-canonical ensemble and the initial temperature was 293 K. The simulation was then restarted five times with each restart calculating a further 20 000 steps. Dur-

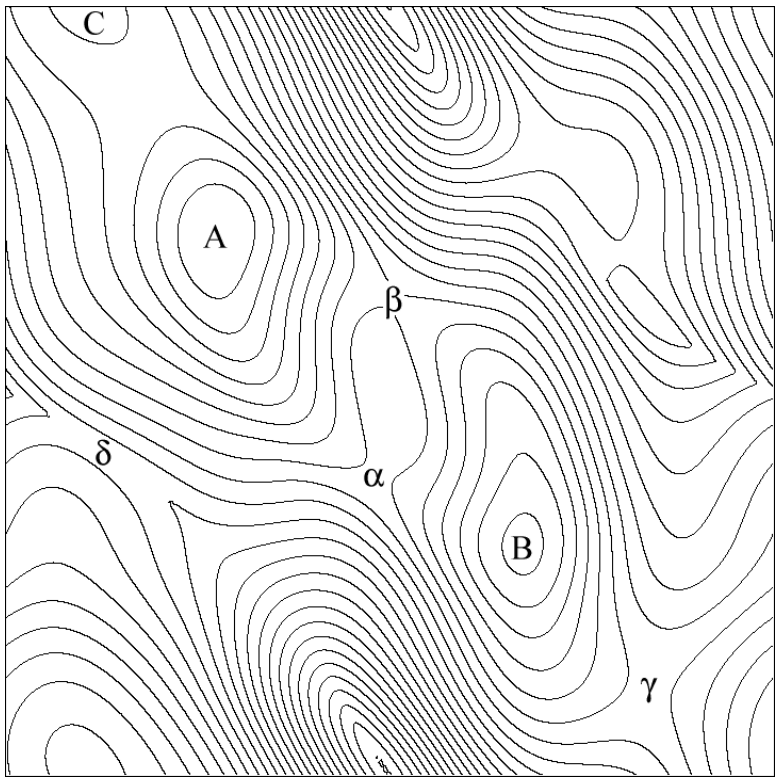

Figure 5.13. Adiabatic potential energy surface of alanine dipeptide with ϕ on the x-axis and ψ on the y-axis. A, B and C are three minima and α, β, γ and δ are transition points.

ing these simulations the dipeptide remained in minimum B. The ends of the last five simulations provided five different starting points for the RDFMD experiments. Each RDFMD simulation was started from all of these five start points. This was to ensure that the results were reproducible and not a result of a particular starting conformation.

To induce conformational change using a digital filter it is necessary to identify the target frequencies. Figure 5.14 shows the spectral density of the second 40 ps simulation and the amplitude spectra obtained from the ϕ and ψ dihedral angle trajectories. The largest amplitude low frequency peak is the ψ dihedral peak at $50\,\mathrm{cm}^{-1}$. This peak was chosen as the initial target for amplification.

To target the ψ dihedral angle, the filter shown as a solid line in figure 5.15 was designed in MATLAB. The design command used was `fircls(2000,[0 0.0054 0.0066 1],[1 2 1],[1.01 3 1.01],[0.99 2 0.99],'text')`. This means that the filter has 2001 coefficients and has transitions at $45\,\mathrm{cm}^{-1}$ and $55\,\mathrm{cm}^{-1}$, moving from a response of 1 to 2 to 1. The response of 1 is constrained to be between 0.99 and 1.01 and the

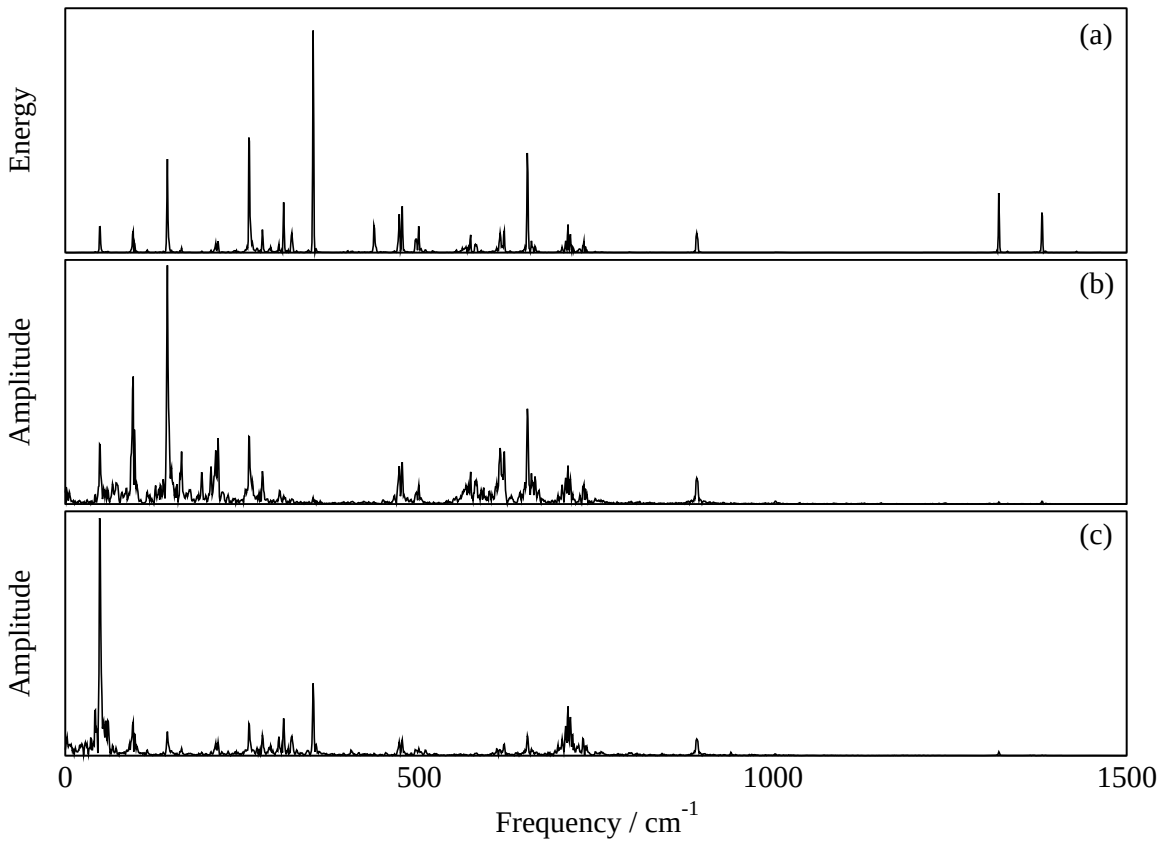

Figure 5.14. Alanine dipeptide: (a) the spectral density of a 40 ps MD simulation; (b) the amplitude spectrum of the ϕ dihedral angle; (c) the amplitude spectrum of the ψ dihedral angle. All the y axes are to different scales.

amplifying response is constrained between 2 and 3. The small number of coefficients used mean that the constraints cannot be perfectly satisfied and the filter amplifies by at most 1.84. The filter response is a long way from the ideal response and is sloped rather than square, but this should not matter because the target frequency is also not precisely defined.

When attempting to use the $50\,\mathrm{cm}^{-1}$ amplifying filter to produce a transition, it was found that after a few filter applications the $50\,\mathrm{cm}^{-1}$ filter did not introduce any more energy into the system, but that the $40\,\mathrm{cm}^{-1}$ amplifying filter shown as a dashed line in figure 5.15 could. The design command for the $40\,\mathrm{cm}^{-1}$ filter was the same as the $50\,\mathrm{cm}^{-1}$ filter but with transitions at $35\,\mathrm{cm}^{-1}$ and $45\,\mathrm{cm}^{-1}$. It was postulated that the reason that the $50\,\mathrm{cm}^{-1}$ filter stopped working and the $40\,\mathrm{cm}^{-1}$ filter was able to add energy, was that the frequency of the target dihedral had shifted. The investigation of the frequency shifting will be presented first, followed by the experiments to cause

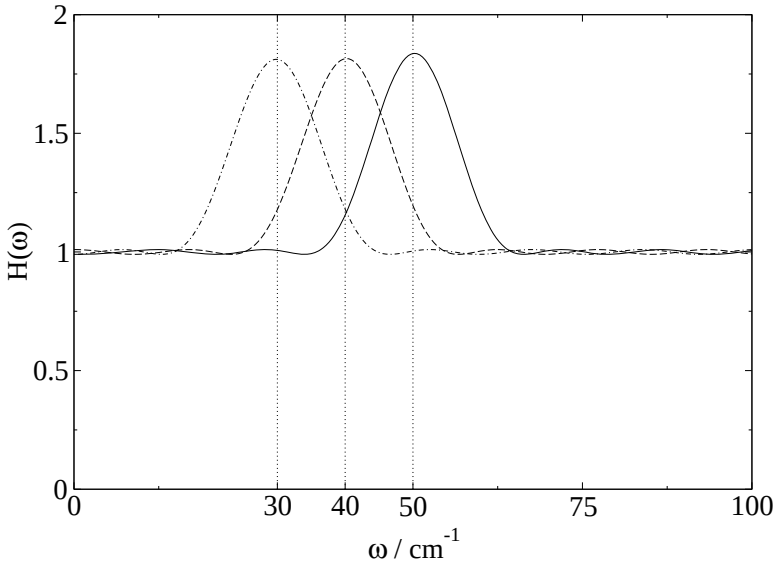

Figure 5.15. Three filters designed in MATLAB for amplifying the ψ backbone dihedral of alanine dipeptide. The filter depicted as a solid line targets $50\,\mathrm{cm}^{-1}$, the dashed filter targets $40\,\mathrm{cm}^{-1}$ and the dot-dash filter targets $30\,\mathrm{cm}^{-1}$. All filters use 2001 coefficients.

Table 5.1. Average energies after each application of the $50\,\mathrm{cm}^{-1}$ amplifying filter.

Filter application number	Energy / kJ mol^{-1}		
	Total	Kinetic	Potential
–	-147.0	11.5	-158.5
1	-147.3	11.4	-158.7
2	-145.5	12.0	-157.5
3	-145.0	12.2	-157.2
4	-144.9	12.3	-157.2
5	-144.9	12.3	-157.2

transitions.

To investigate the frequency shift, one of the start points was chosen and the $50\,\mathrm{cm}^{-1}$ filter was applied five times with a zero step delay between applications. As with the pentane simulations, the zero step delay ensures that any effect is caused by the nature of the filter applied rather than by the time of application. Table 5.1 lists the average energies for each 2001 step simulation. The table shows a small amount of energy being put in by the first four applications but no change after the 5th filter application.

Continuing from the second application of the $50\,\mathrm{cm}^{-1}$ filter, the $40\,\mathrm{cm}^{-1}$ amplifying filter was applied three times. The resulting average energies are listed in table 5.2. The energies quickly increase with the $40\,\mathrm{cm}^{-1}$ filter and after the third application a

Table 5.2. Average energies after each application of the $40\,\mathrm{cm}^{-1}$ amplifying filter.

Filter application number	Energy / $\mathrm{kJ\,mol}^{-1}$		
	Total	Kinetic	Potential
–	-145.5	12.0	-157.5
1	-138.4	15.4	-153.8
2	-129.0	19.5	-148.5
3	-120.5	25.9	-146.4

conformational transition is produced via transition state γ. These simulations are a clear example of an important motion changing frequency as energy is added slowly.

The protocol used to cause conformational change was as follows:

1. Apply the $50\,\mathrm{cm}^{-1}$ targeting filter with a pause of 15 steps between applications until it is no longer effective.
2. Continue from the last effective application with the $40\,\mathrm{cm}^{-1}$ targeting filter until it is no longer effective or conformational change has been caused.
3. Continue from the last effective application with the $30\,\mathrm{cm}^{-1}$ targeting filter until it is no longer effective or conformational change has been caused.

The $30\,\mathrm{cm}^{-1}$ targeting filter is shown in figure 5.15. Its design equation is the same as the other two filters except that the transition boundaries are at $25\,\mathrm{cm}^{-1}$ and $35\,\mathrm{cm}^{-1}$. The delay of 15 steps was chosen to allow the mid-point of the filter application to move each time. This is desirable because the mid-point of the filter application may be an unfortunate conformation with high potential energy in the mode being targeted. In this worst case the filter will do very little, but by moving the application point slightly each time, subsequent applications will do more. Having the mid-point move slightly rather than a great deal means that these simulations are more similar to solution-phase simulations where energy dissipation becomes important.

The protocol given above was applied to the five start points. Table 5.3 lists the number of applications of each filter needed to cause conformational change. In no cases was it necessary to use the $30\,\mathrm{cm}^{-1}$ targeting filter, but a second set of simulations were run from each of start points 1 and 2 in which the $30\,\mathrm{cm}^{-1}$ filter was used directly after the $50\,\mathrm{cm}^{-1}$ targeting filter. These two sets of simulations, listed at the bottom of table 5.3, required fewer filter pulses than the corresponding simulations that used the $40\,\mathrm{cm}^{-1}$ filter, suggesting that the frequency of the dihedral motion has shifted closer to $30\,\mathrm{cm}^{-1}$

Table 5.3. The number of filter applications needed to produce conformational change in alanine dipeptide. No conformational change was found in the simulations beginning with start point 3.

Start Point	Number of Applications of Filter			
	$50\,\mathrm{cm}^{-1}$	$40\,\mathrm{cm}^{-1}$	$30\,\mathrm{cm}^{-1}$	Total
1	5	4	–	9
2	3	4	–	7
3	18 . . .	–	–	> 18
4	9	3	–	12
5	6	3	–	9
1	5	–	1	6
2	3	–	3	6

than $40\,\mathrm{cm}^{-1}$ in at least these cases. The simulation set beginning with start point 3 behaved differently to the others. Even after 18 applications of the $50\,\mathrm{cm}^{-1}$ targeting filter, the dihedral motion was still only increasing slightly. Undoubtedly, conformational change could have been achieved by switching to one of the lower frequency filters but this was an interesting result in itself.

Figure 5.16 shows the ϕ-ψ trajectory from start point number 1, using the $50\,\mathrm{cm}^{-1}$ and $40\,\mathrm{cm}^{-1}$ filters. The final transition is from minimum B to A via transition state γ. The transitions from start points 2, 4 and 5 were very similar.

Targeting the ψ backbone dihedral angle is clearly an effective way of promoting a conformational transition from B to A. Is it possible to cause conformational change while targeting the ϕ angle though? Figure 5.17 shows the low frequency part of figure 5.14(b) which is the amplitude spectrum obtained from a ϕ angle trajectory. Overlayed on this is the filter designed to preferentially target the ϕ angle. The MATLAB design command for this filter was `fircls(2000,[0 0.0144 0.0192 1],[1 2 1],[1.01 3 1.01],[0.99 2 0.99],'text')` which is the same as the previous filters used on alanine dipeptide but with transitions at $120\,\mathrm{cm}^{-1}$ and $160\,\mathrm{cm}^{-1}$.

Start point number 5 was chosen for this part of the study as the motions in the ϕ and ψ dihedral angles were naturally very similar in magnitude. The filter was used with a delay of 15 time-steps between applications. The resulting trajectories are shown in figure 5.18. Eight filter applications were necessary to produce conformational change and, though the transition is once again via transition point γ, the trajectory before the

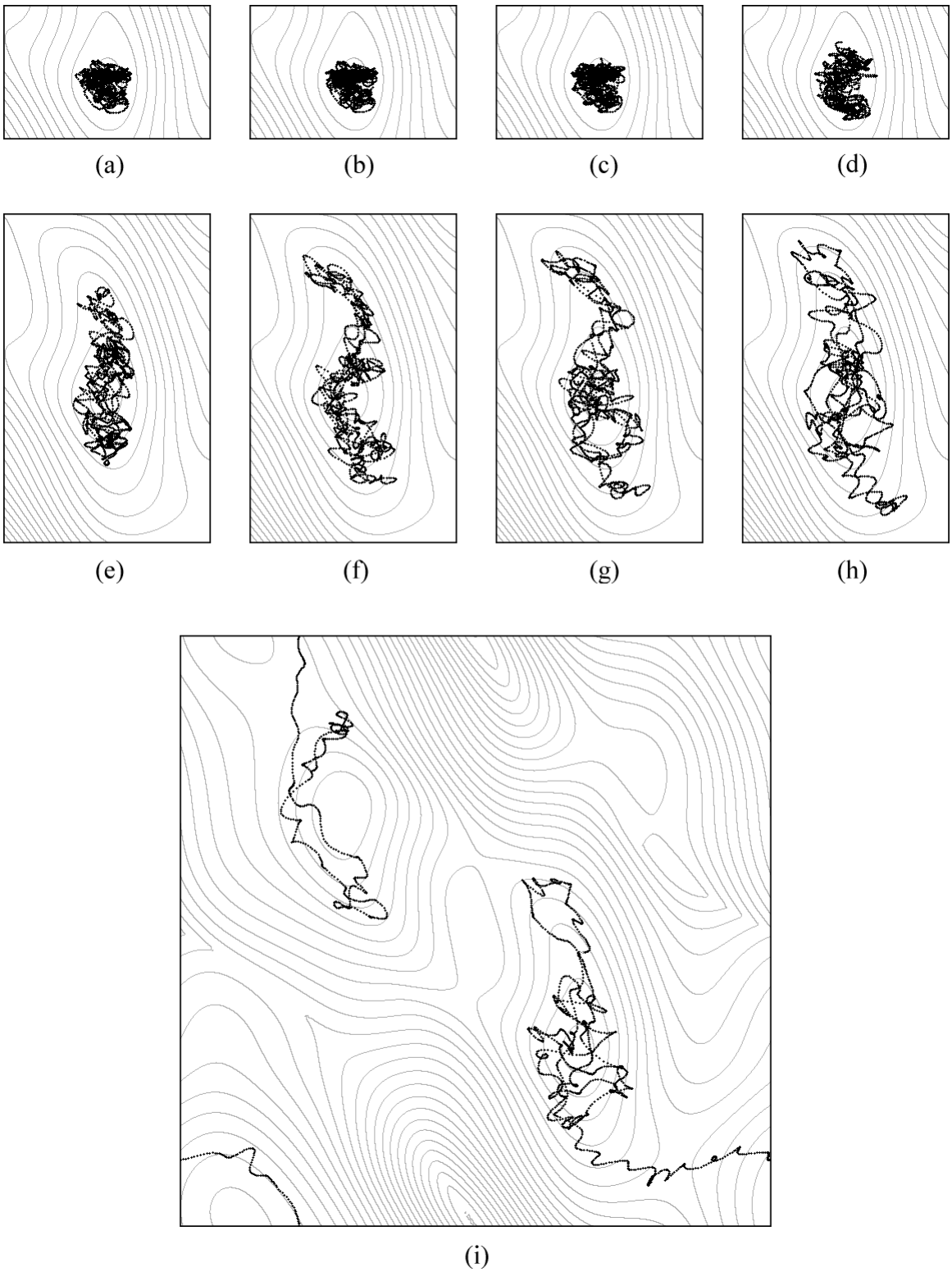

Figure 5.16. The ϕ-ψ trajectory of alanine dipeptide resulting from RDFMD filter applications targeting the ψ dihedral angle. All parts show ϕ on the x-axis and ψ on the y-axis. Parts (a)–(h) show just small portions of the surface around minimum B. Part (i) shows the whole surface from -180°–180° in both axes. Parts (a)–(e) show the result of successive applications of the $50\,\mathrm{cm}^{-1}$ targeting filter and parts (f)–(i) result from continuing the sequence with the $40\,\mathrm{cm}^{-1}$ targeting filter.

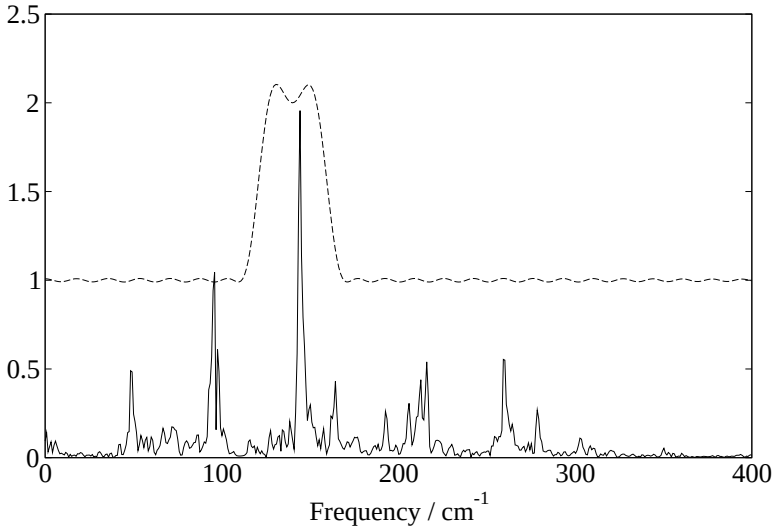

Figure 5.17. The solid line is the low frequency part of the amplitude spectrum obtained from a 40 ps ϕ dihedral angle trajectory. The dashed line is the frequency response of the filter designed to target the ϕ dihedral. The data in the dihedral angle spectrum has been scaled in the y-axis to fit on the same graph as the filter response.

transition is very different to those produced by the filters targeting ψ (compare with figure 5.16 for instance). The trajectory immediately before the final filter application shows the ϕ angle with a larger range of motion than the ψ angle—contrary to what would be expected from the shape of the potential energy surface although consistent with the digital filter being applied. This result shows once again the specificity of the method.

5.4.3: Summary

The study of gas-phase alanine dipeptide has shown three things:

1. The frequency of a target motion can change as energy is added to the system. The RDFMD technique can adapt to these changes in frequency.
2. RDFMD can be used to produce conformational change in this system from a variety of start points.
3. By using different filters it is possible to target either the ϕ or ψ dihedral angles preferentially.

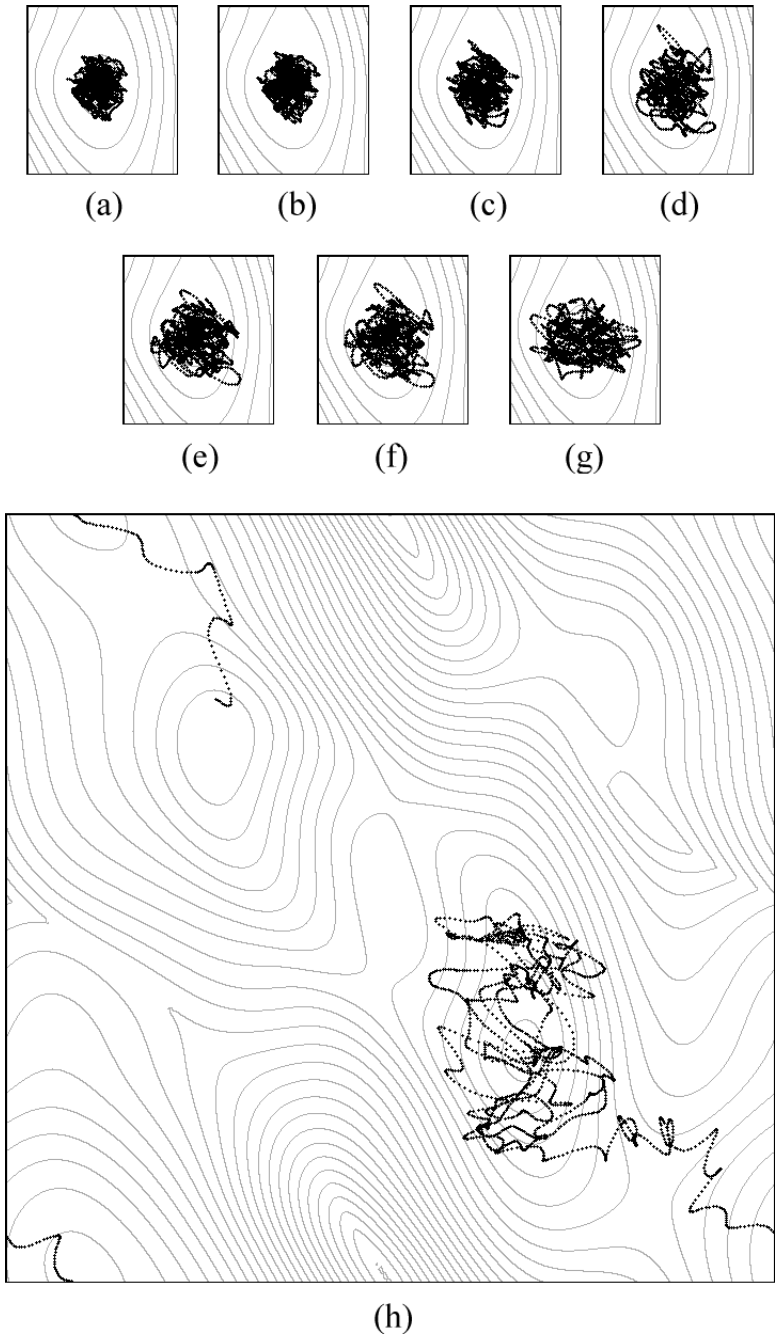

Figure 5.18. The ϕ-ψ trajectory of alanine dipeptide resulting from RDFMD filter applications targeting the ϕ dihedral angle. All parts show ϕ on the x-axis and ψ on the y-axis. Parts (a)–(g) show just small portions of the surface around minimum B. Part (h) shows the whole surface from -180°–180° in both axes. All parts of the trajectory are shown; eight filter applications were needed to produce conformational change.

5.5: Solvated Alanine Dipeptide

The next step in increasing the complexity of the system being studied, is to look at

solvated alanine dipeptide. RDFMD should be able to put energy into a solvated system gradually, overcoming the energy dissipation problems of DFMD.

Water was the first solvent to be tried. The alanine dipeptide molecule was solvated in a cubic box of side 33.3 Å containing 1145 TIP3P (ref. 88) molecules using a version of BOSS (ref. 6) modified to produce cubic solvent boxes. A Monte Carlo (MC) simulation of 1 500 000 steps at constant volume was performed to remove any bad interactions. The final configuration of the MC simulation was converted into a coordinate input file for DLPROTEIN (ref. 19) and the extra force field parameters for water and water-solute interactions were added to the DLPROTEIN force field file used for the gas-phase simulations.

In the molecular dynamics simulations, the smooth particle mesh Ewald[24] (SPME) method was used for the electrostatic component of the force calculation, and was parametrized according to the protocol given in the DLPROTEIN manual. Cut-offs were set to 8 Å, all bonds were constrained by SHAKE (ref. 8) and a 2 fs time-step was employed. An initial simulation of 50 000 steps was performed in the NVT (canonical) ensemble at a temperature of 293 K using the Nosé-Hoover thermostat[102] with a time constant of 0.5 ps. Figure 5.19 plots the ϕ-ψ trajectory of this simulation. It is clear from the figure that the potential energy minima of alanine dipeptide in aqueous solution are in different positions to the gas-phase simulations. This result has also been observed both experimentally[103] and theoretically[98] and is not surprising because the van der Waals and hydrogen bond interactions between the peptide and the polar solvent are not insignificant. However, it does mean that to examine any conformational change properly, a new potential energy surface would need to be calculated. Generating the ϕ-ψ free-energy surface for this system would be an expensive calculation and was not therefore attempted. Instead, an alternative solvent was tried.

Water was unsuitable because it was too polar. The second solvent tried was chloroform ($CHCl_3$) which, it was hoped, would still cause energy dissipation, but interact with the dipeptide sufficiently weakly so that the positions of the potential minima remained the same. The intermolecular potential for chloroform used was the default BOSS potential.[104] The equilibrium bond lengths and angles were also taken from BOSS. The valence angle potential for chloroform was taken from a model of carbon tetrachloride.[105]

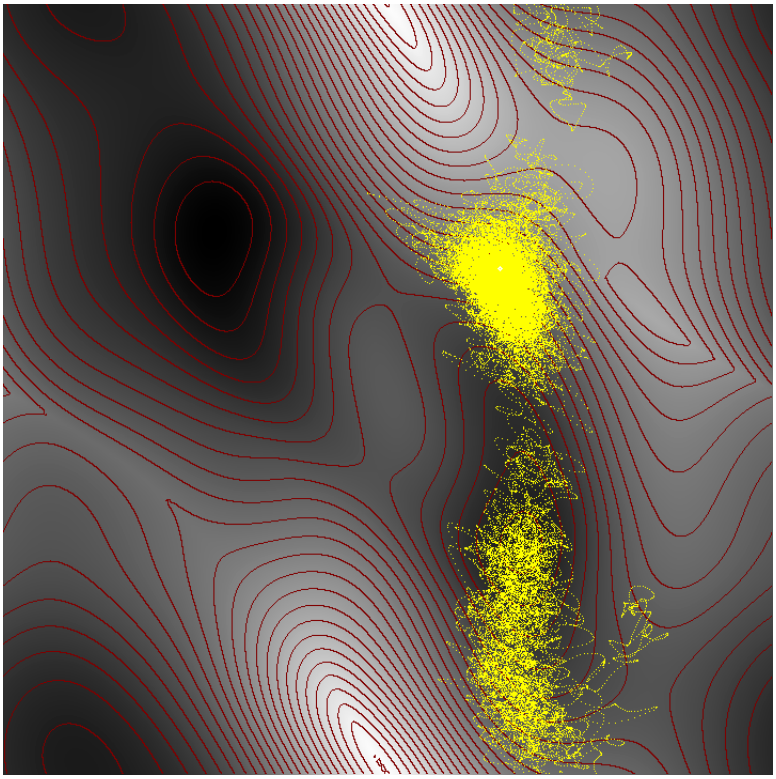

Figure 5.19. ϕ-ψ trajectory of alanine dipeptide in water plotted on the adiabatic potential energy surface. This demonstrates that the minima for this system are significantly different to the gas-phase.

The molecular dynamics system was set up by solvating the dipeptide in a cubic box of side 33 Å containing 310 chloroform molecules, using the modified version of BOSS and the same minimisation protocol used for water above. The SPME was re-parametrized for the new system and 8 Å cut-offs were once again used. SHAKE was used to constrain all bonds and a 2 fs time-step was employed. A 100 000 step MD simulation was performed in the NPT (isobaric-isothermal) ensemble at 220 K and 1 atm pressure using the Melchionna[106] adaptation of the Hoover barostat and thermostat. The low temperature was to prevent the dipeptide from spontaneously changing conformation and is 10 K higher than the freezing point of chloroform (210 K). During the 200 ps simulation, the temperature and pressure settled down and the dipeptide remained in minimum B. This suggested that the gas-phase minima and the minima of the dipeptide solvated in chloroform were very similar and that the adiabatic potential energy surface could serve to highlight conformational change events.

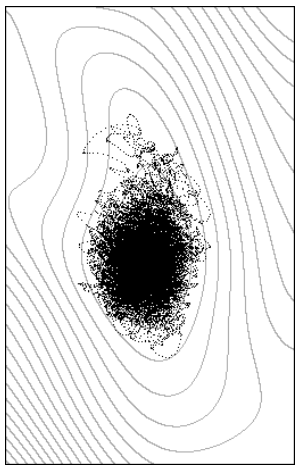

Figure 5.20. The ϕ-ψ trajectory of alanine dipeptide solvated in chloroform during a 66 ps NVE simulation. Just the part of the potential energy surface around minimum B is shown.

5.5.1: Filter Design

The data used to determine the design of the filter was obtained from a 33 000 step NVE simulation restarted from the 200 ps NPT simulation above. 33 000 steps gives a frequency resolution of $0.5\,\mathrm{cm}^{-1}$ which is more than sufficient. The ϕ-ψ trajectory taken by the simulation is shown in figure 5.20. Even at 220 K, the ranges of the dihedral motions are quite large. This is caused by the frequent collisions of the solvent with the solute.

Figure 5.21 shows the spectral density of the simulation and the ϕ and ψ amplitude spectra. Part (b) shows that the ϕ dihedral angle has large amplitude motion across a broad range of frequencies, whereas part (c) indicates that the large amplitude motion in the ψ dihedral is almost completely confined to the 0–100 cm^{-1} region.

To examine the distribution of the motion with frequency more closely, the trajectories of all the internal degrees of freedom (valence angles and dihedrals) were calculated from the Cartesian coordinates. There are fifteen valence angles and four sets of four dihedral angles. For instance, the ψ dihedral is generally defined to be the dihedral angle made by the four backbone atoms (C–N–CA–C in this case), but this dihedral makes a set of four along with the other three dihedrals sharing the same two central atoms (C–N–CA–CB, HN–N–CA–C and HN–N–CA–CB). The four dihedral angles in a set can be expected to be closely related.

The manner in which the standard deviation of each internal degree of freedom

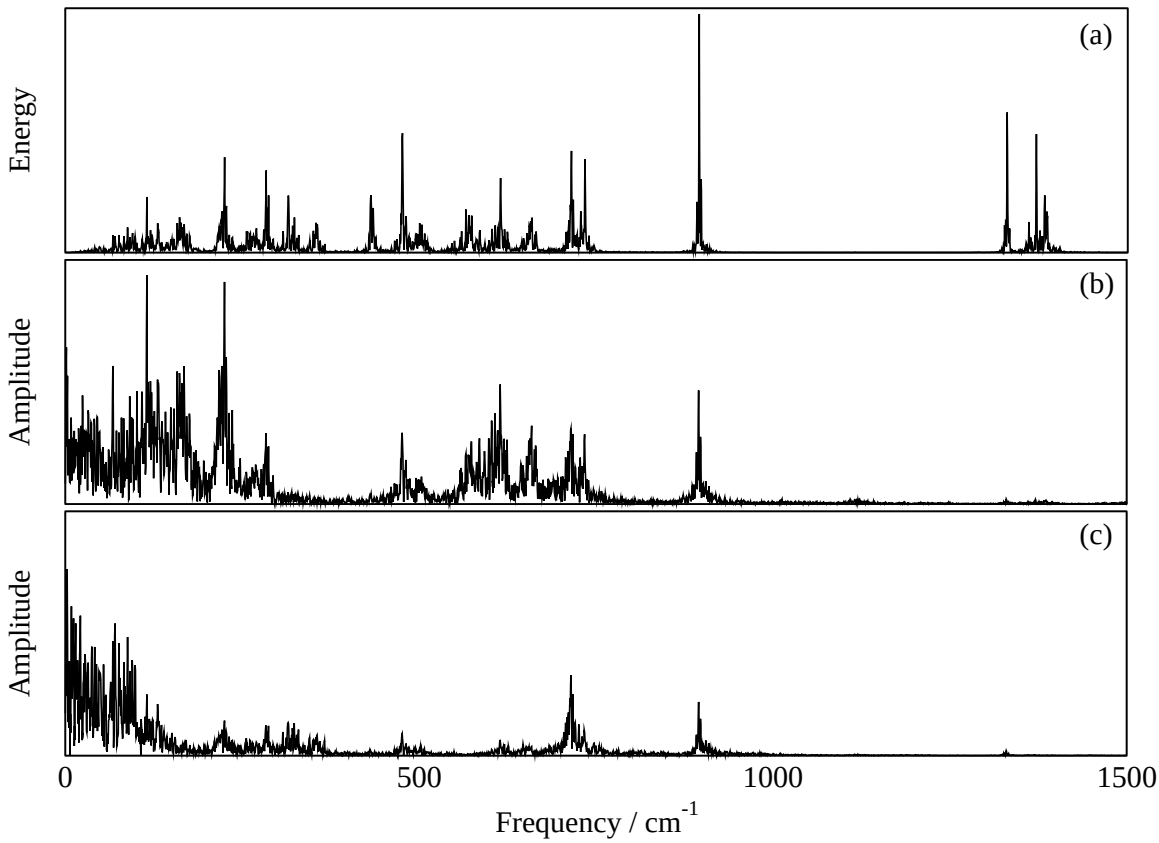

Figure 5.21. Alanine dipeptide in chloroform: (a) the spectral density of a 66 ps NVE MD simulation; (b) the amplitude spectrum of the ϕ dihedral angle; (c) the amplitude spectrum of the ψ dihedral angle. All the y-axes are on different scales. Please see the text for a caveat on comparing the spectra.

trajectory changes as the high frequency components of the motion are removed will help show the frequency with which each degree of freedom moves. To do this, each trajectory initially has its mean subtracted. The trajectories are then Fourier transformed into the frequency domain, and all data above a cut-off frequency is set to zero. The signals are then back-transformed to the time domain and their standard deviations calculated. The standard deviations are then measured, percentage-wise, against the standard deviations of the original signals. Nine frequency cut-offs were chosen to lie (as far as possible) between major peaks in the spectral density, so as to provide information on the type of motion represented by each peak or region.

Figure 5.22 plots the percentage of the original standard deviation for each of the 31 degrees of freedom and each of the nine frequency cut-offs. The percentage standard deviation for the thirteen of the fifteen valence angles is less than 25% by 400 cm^{-1}. This shows that most of the motion in the valence angles is at frequencies greater

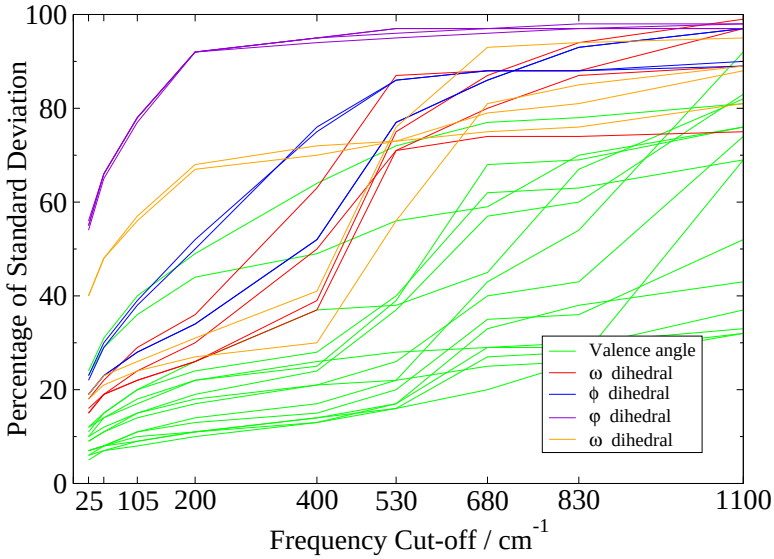

Figure 5.22. Each point in the plot corresponds to a particular degree of freedom and a particular frequency cut-off. The chosen cut-offs are marked on the x-axis, including a cut-off at $50\,\mathrm{cm}^{-1}$. The standard deviation of the original signal is compared with the standard deviation of the signal after passing it through a frequency filter with a high-frequency cut-off. This gives the percentage of standard deviation plotted on the y-axis.

then $400\,\mathrm{cm}^{-1}$. The set of four ψ dihedrals are almost indistinguishable in their attenuation and have greater then 50% standard deviation even with the $25\,\mathrm{cm}^{-1}$ cut-off, demonstrating that a large proportion of the motion in the ψ dihedral is at frequencies less than $25\,\mathrm{cm}^{-1}$. The next highest signals at $25\,\mathrm{cm}^{-1}$ are, surprisingly, the dihedrals HN–NT–C–CA and HN–NT–C–O. These dihedrals are members of the ω set next to the ψ dihedral (see figure 5.10). They have motion at this low frequency because of a small motion in ψ away from the minimum, approximately 30 000 steps into the simulation (this deviation from the normal motion can be seen in figure 5.20 as unusually high ψ values). The motion was transfered into the neighbouring ω dihedral and caused the hydrogen atom to vibrate at a similar low frequency. The four dihedrals in the ϕ set have around 20% standard deviation by the $25\,\mathrm{cm}^{-1}$ cut-off point. This is consistent with the earlier analysis that the energy in the ϕ dihedral is more dispersed than the ψ dihedral. The two valence angles that have significantly higher percentage standard deviations than the others are N–CA–C and CB–CA–C. These angles are involved in the ψ dihedral motion.

Taking all this analysis into consideration, it was decided that the best filters for

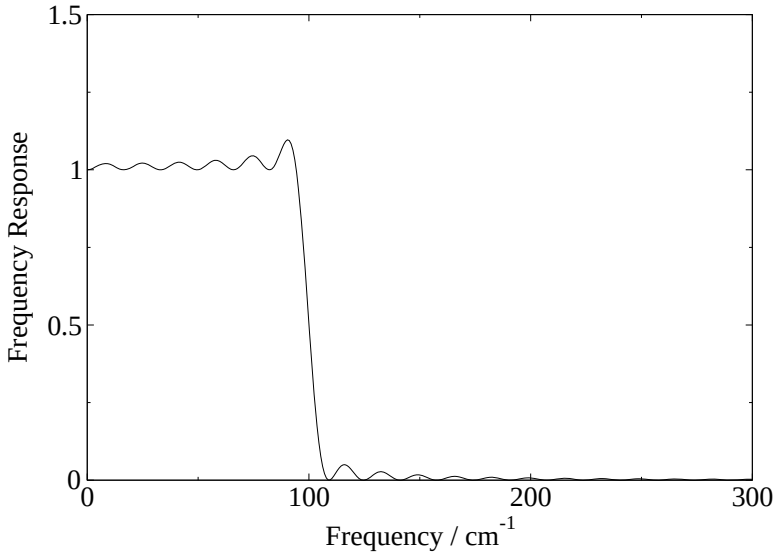

Figure 5.23. The MATLAB design command for this filter is `fircls(2000,[0 0.0120 1],[1 0],[1.1 0.05],[1 0],'text')`. It has 2001 coefficients and a transition from 1 to 0 at $100\,\mathrm{cm}^{-1}$. The response is constrained to lie within 1–1.1 and 0–0.05.

inducing conformational change would target only the low frequency region. A filter targeting 0–100 cm^{-1} will capture approximately 75% of the ψ motion and almost 40% of the ϕ motion (according to the standard deviation) while not affecting the other degrees of freedom significantly. All the filters used in the rest of this section are derived from the filter shown in figure 5.23 using the first two filter properties discussed in section 3.3.3. To create a filter which amplifies 0–100 cm^{-1} by three and the rest by one, the filter of figure 5.23 (which has responses of one and zero in the two regions) is multiplied by two (to yield a filter with responses of two and zero) and one is added to its middle coefficient to raise the magnitude of the whole frequency response by one and yield the required filter.

5.5.2: Energy Dissipation

Before any attempt is made to induce conformational change in the system, several start points must be generated (as was done for the gas-phase simulations) and a sensible number of steps to wait between filter applications must be determined. Five start points were generated by running five successive 32 ps NVE simulations with the first restarting from the 66 ps NVE simulation described earlier.

To find a good value for the time-step delay between filter applications, the rate of energy dissipation from the internal velocities of the alanine dipeptide was investigated (where "internal velocities" refers to the atomic velocities with the rotational and translational components removed, as used to fill the filter buffer). To do this, a filter amplifying 0–100 cm^{-1} by 20 times was applied to each of the five start points. The kinetic energy of the internal velocities was calculated at each step and first order decay curves were fitted to the resulting data. The five start points provide ten possible energy decay measurements by taking data from both forwards and backwards simulations. The third start point was not affected by the filter application and so an exponential decay curve could not sensibly be fitted. The remaining eight half-life measurements were 11, 11, 12, 15, 18, 21, 21 and 30 time-steps. In a condensed-phase simulation such as this, we therefore need a delay that is long enough to allow the filter mid-point to move, but short enough such that the energy introduced by one filter application has not dissipated before the time of the following filter application. Fifteen time-steps was used in the gas-phase simulations and fifteen was also chosen for the following solution-phase simulations.

5.5.3: Inducing Conformational Change

Conformational change was produced in all five test cases using the following protocol:

1. Apply a 0–100 cm^{-1} $\times 2$ filter with a 15 time-step delay until conformational change is observed.
2. Take a point a few filter applications before conformational change was caused and apply a 0–100 cm^{-1} $\times 1.5$ filter until conformational change is observed.

The reason for the two-step protocol is that a $\times 2$ filter was found, in some cases, to cause transitions in which the dihedrals were moving quickly, or via high energy points in the potential energy surface. By switching to a $\times 1.5$ filter a few filter applications before the conformational change was observed, the quality of the transition may be improved.

Table 5.4 shows the number of applications of each filter required to produce conformational change, the transition point crossed and the minimum the system settled in after the transition. Some of the trajectories that are listed as finishing in minimum

Table 5.4. The number of filter applications needed to produce conformational change in alanine dipeptide in chloroform. The transition point and minima labels refer to figure 5.13.

Start Point	Filter Applications ×2	×1.5	Total	Transition Point	Final Minimum
1	13	–	13	–	B
1	11	2	13	β	A
2	6	–	6	γ	C
2	4	5	9	γ	C
3	9	–	9	–	B
3	6	5	11	β	A
4	8	–	8	γ	C
4	7	1	8	γ	C
5	10	–	10	–	A
5	7	2	9	β	A

C may, given more simulation time, have moved into minimum A. Where no transition point is listed, the trajectory followed a path over a high energy part of the potential surface for which there is no label. Figure 5.24 shows the two final trajectories originating from start point 1. The first trajectory results from 13 applications of the ×2 filter. The ψ angle in this trajectory spins right round through 360° and the system settles back into the original minimum. The second trajectory is the result of adding the energy more gradually—11 applications of the ×2 filter and 2 applications of the ×1.5 filter. This produces a clean transition over transition state β into minimum A.

To investigate the system further, ten further start points were generated from start point 5 by running ten 8 ps NVE simulations. The same protocol was employed, except that the ×1.5 filter was not used if the transition caused by the ×2 filter was acceptable. The results are listed in table 5.5. It was possible to produce conformational change in nine of these ten start points. Of the nine successful cases, the paths taken included all possible transition points (α, β, and γ) and the dipeptide finished in both minimum C and minimum A (though some may have moved from C to A given more simulation time). It was not possible with this protocol to produce reasonable conformational change from start point 9.

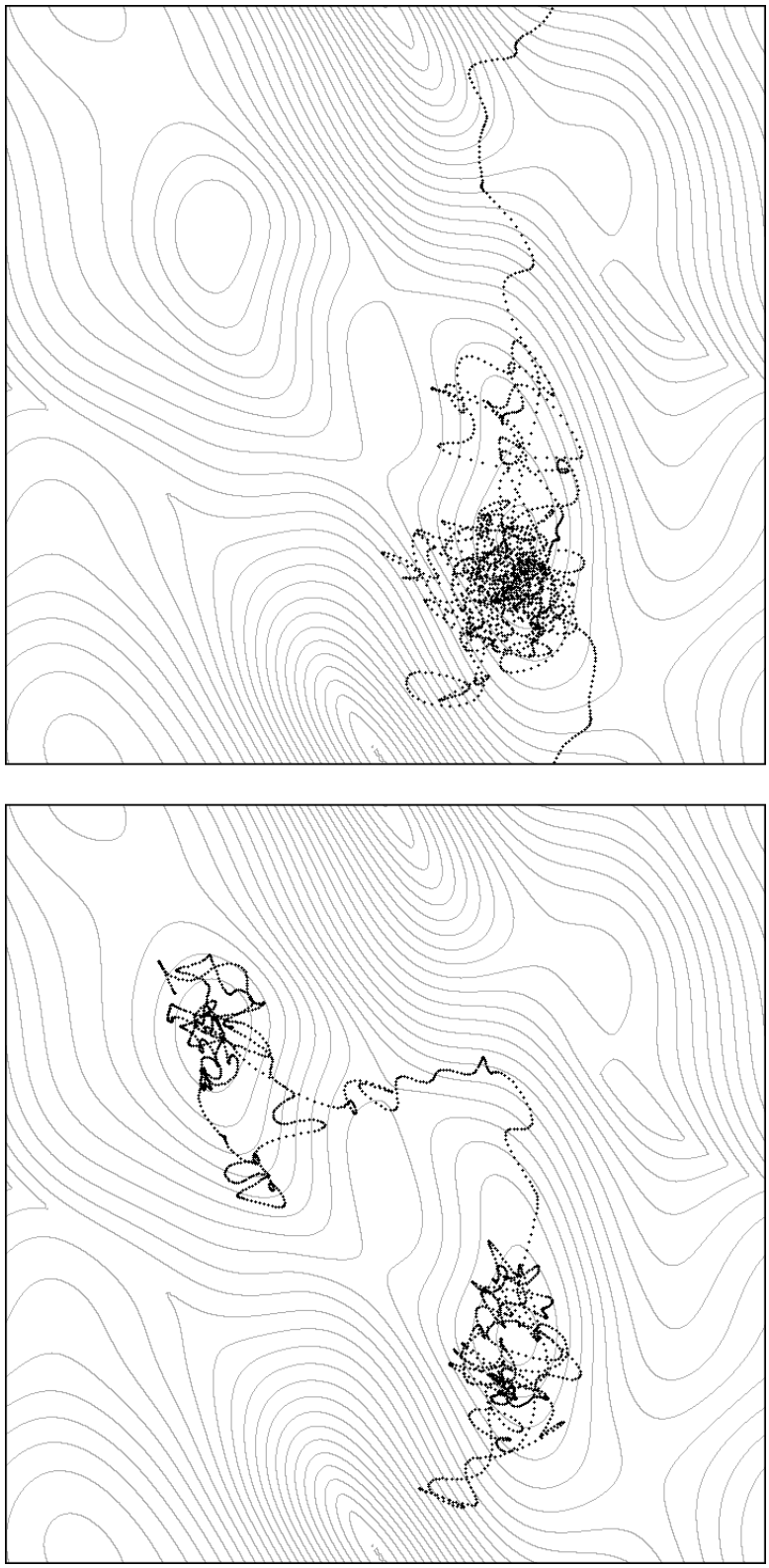

Figure 5.24. Transition trajectories originating from start point 1: (top) the result of applying 13 $\times 2$ filters; (bottom) the result of applying 11 $\times 2$ followed by 2 $\times 1.5$ filters.

Table 5.5. The number of filter applications needed to produce conformational change in alanine dipeptide in chloroform. The transition point and minima labels refer to figure 5.13.

Start Point	Filter Applications ×2	×1.5	Total	Transition Point	Final Minimum
6	6	–	6	γ	C
6	5	2	7	γ	C
7	10	–	10	γ	C
8	13	–	13	β	A
9	5	–	5	–	A
9	2	6	8	–	B
10	15	–	15	γ	C
10	13	4	17	γ	C
11	9	–	9	γ	C
11	8	2	10	γ	C
12	8	–	8	γ	C
12	6	5	11	γ	C
13	4	–	4	β	A
14	8	–	8	γ	C
14	6	3	9	γ	C
15	5	–	5	α	A
15	4	1	5	α	A

5.5.4: Comparison with a Heating Pulse

As a further validation of the technique, the transition trajectories of the final ten start points were compared with trajectories resulting from single-pulse filter applications, where the frequency response of the filter was the same for all frequencies. Application of one of these filters scales all frequencies to the same degree and can be called a "heating pulse". For each of the ten start points, the mean total energy of the best transition was measured. A heating filter was designed (by trial and error) to reproduce this total energy in a single application. Each heating pulse was tuned to give at least the same energy as the best transition and no more than $50\,\mathrm{kJ\,mol^{-1}}$ (approximately 1% of the total) higher. No transition were induced by the heating pulses, once again demonstrating the benefit of the frequency selective nature of the method.

5.5.5: Summary

The frequency spectrum of alanine dipeptide solvated in chloroform was studied by

spectral density, amplitude spectra of the ϕ and ψ angles, and by measuring the change in the motion of every internal degree of freedom as a series of high-frequency cut-offs were applied. This analysis led to the use of filters amplifying the 0–100 cm^{-1} region for promoting conformational change. The half-life of the energy dissipation from the internal velocities of alanine dipeptide was measured and a 15 time-step delay between filter pulses was chosen as a result. A protocol using $\times 2$ filters followed by $\times 1.5$ filters was designed and applied to fifteen start points. It was possible to produce good conformational transitions in all but one of the start points via a variety of routes. The application of a measured heating pulse to ten of the start points produced no conformational change, demonstrating the frequency selectivity of the digital filters.

5.6: Peptide YPGDV

The final and most complex system studied by RDFMD was the pentapeptide YPGDV (Tyr-Pro-Gly-Asp-Val) solvated in water. NOE data for this system was experimentally determined by Dyson *et al.*[107] in 1988. The *trans* form of the peptide (shown in figure 5.25) was found to form a high population (approximately 50%) of reverse turns in water at 278 K. Tobias, Mertz and Brooks[108] performed two 2.2 ns MD simulations of YPGDV at 300 K in 1991 using the CHARMM peptide parameters[5] and TIP3P water.[88] They started one simulation in an extended conformation and another simulation in a type II reverse turn conformation, and found that the extended conformation folded and the folded conformation became extended. The two trajectories were further analysed in 1993 by Karpen, Tobias and Brooks[109] using the ART-2′ clustering algorithm, which is based on a self-organising neural network and is implemented in the CHARMM molecular mechanics program.[5]

Wu and Wang[60] recently simulated YPGDV using normal MD and the SGMD method (see section 2.2.3). The simulations were performed in CHARMM using the CHARMM22 force field[110] and TIP3P water. A 2 ns MD simulation initiated from an extended conformation did not fold into a type II turn, whereas a 2 ns SGMD simulation started from the same extended conformation sampled more conformations, including a type II turn. The conformations sampled in the SGMD simulation were clustered according to Wu and Wang's own method (see below) and the conformation at the centre

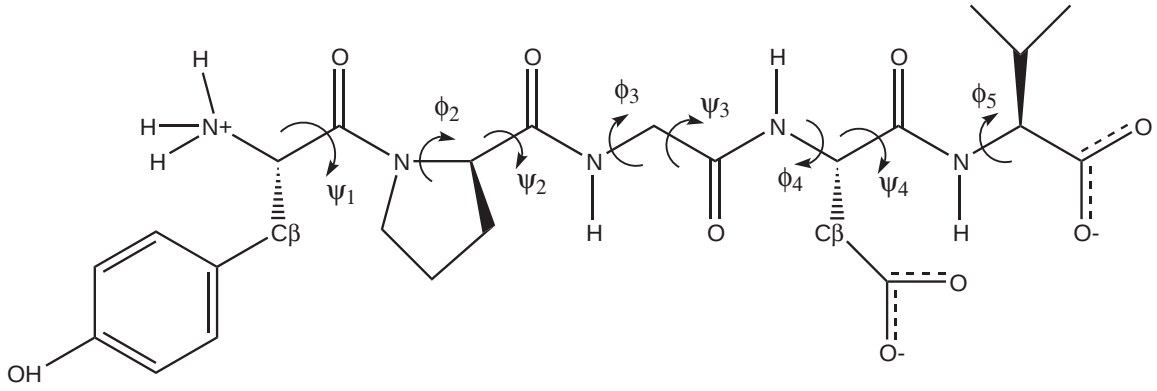

Figure 5.25. Tyr-Pro-Gly-Asp-Val (or YPGDV). The four ϕ and four ψ dihedral angles are shown (numbered according to residue).

of the largest cluster was used as the start point for a further 2 ns normal MD simulation, during which no conformational change was observed. The implication is that the conformation at the centre of the largest cluster is in a very stable conformation. Two of the methods used in the paper—conformational clustering and calculation of NOEs—have been used to analyse the RDFMD simulations of the pentapeptide and are described below.

5.6.1: Conformational Clustering

The aim of conformational clustering is to extract from an MD (or MC) trajectory, succinct and useful information about the conformations sampled. Clustering algorithms in general, analyse a set of configurations, cluster similar configurations together about cluster centres, and report the cluster centres along with the sizes of their respective clusters. The cluster centres should represent all the conformational diversity present in the original trajectory.

The clustering algorithm of Wu and Wang[60] firstly defines a "similarity function" for conformations. The similarity function is used to measure how close two conformations are, and is defined in terms of the backbone ϕ and ψ dihedrals. For each backbone dihedral θ_i (representing ϕ or ψ), the similarity between two conformations, m and n, is:

$$s_i(m,n) = \frac{C}{C + [\theta_i(n) - \theta_i(m)]^2} \tag{5.2}$$

where C is a constant (set to 300 degree^2 in ref. 60). Angle periodicity is taken into

account when finding the difference of the two dihedrals.

For a peptide with a total of M backbone ϕ and ψ dihedrals, the similarity function, S, between m and n for the entire molecule is simply the product of all the s_i terms:

$$S(m,n) = \prod_{i=1}^{M} s_i(m,n) \tag{5.3}$$

$S(m,n)$ is unity when m and n are the same and approaches zero when they are very different. A high value of $S(m,n)$ means that m and n are very close in conformational space (i.e. the M-dimensional space of the backbone dihedrals).

Finally, the "similarity density" $\rho(k)$ of each conformation k, is calculated:

$$\rho(k) = \frac{1}{N} \sum_{n=1}^{N} S(k,n) \tag{5.4}$$

where N is the total number of conformations being clustered. The similarity density of conformation k is a measure of the number of conformations similar to k. A large value of $\rho(k)$ means that there are many conformations similar to conformation k.

A cluster is defined as a conformation with a locally maximum similarity density. A cluster resolution, S_0, is needed to properly define "locally maximum". A cluster centre is formally defined as follows:

> A cluster centre, c, must satisfy $\rho(c) \geq \rho(i)$ for all conformations i which have $S(i,c) > S_0$.

That is, a conformation that has more conformations similar to it than any other conformation that is sufficiently similar to it.

Once the cluster centres have been found, each of the remaining conformations is assigned to the cluster centre to which it is the most similar. Or, formally:

> A conformation i is assigned to cluster centre c if the similarity function $S(i,c)$ is greater than the similarity function for i with any other cluster centre, provided that $\rho(i) < \rho(c)$.

For Wu and Wang's simulations of YPGDV, C was set to be 300 degree^2 and the cluster resolution, S_0, was defined to be $(3/4)^8$ (the eight arising from the eight dihedral angles being considered). This choice is somewhat arbitrary and means that if, for a

pair of conformations m and n, each dihedral angle differs by 10° then the conformations will be sufficiently different (that is $S(m,n) < S_0$) that both could be cluster centres. In the other extreme, if all the dihedral angles between m and n were the same apart from one, then that remaining dihedral angle would have to be more than 51.9° different for the conformations to be different cluster centres.

There are two main problems with this method:

1. In a poly-peptide such as YPGDV, a small change in one of the central backbone dihedral angles leads to a large overall conformational change, relative to a small change in one of the dihedral angles on the end of the chain. The method does not distinguish between these events.
2. Two parameters are required (C and S_0). It is difficult to rationalise the choice of these parameters, and the number of clusters the algorithm finds is sensitive to the parameter choice.

The main use of the clustering algorithm in the work presented here, has been as part of a test for conformational change applied during the RDFMD simulations.

5.6.2: Estimation of NOE Intensities

In the 1988 work of Dyson *et al.*[107], more than forty linear peptides in water were studied by ^{1}H nuclear magnetic resonance experiments. The 300 MHz ROESY spectrum of YPGDV shows an unusually strong NOE between the Gly3 and Asp4 amide protons ($d_{\mathrm{NN}}(3,4)$) in the *trans* form of the peptide and also a cross peak between the $C^{\alpha}H$ resonance of Pro2 and the NH resonance of Asp4 ($d_{\alpha\mathrm{N}}(2,4)$). These connectivities "provide unequivocal evidence for a relatively high population of a $4 \rightarrow 1$ hydrogen-bonded β-turn in YPGDV." This conclusion is also confirmed by circular dichroism studies. Dyson *et al.* estimated that the proportion of β-turn conformations for YPGDV was approximately 50%. No evidence was found for a β-turn in the *cis* form of YPGDV.

Wu and Wang estimated the NOE intensities for several proton pairs and compared their estimation to the experimental result. The method of estimation used by Wu and Wang and also in this work is presented here.

The NOE represents average structural information and depends on the sixth power

of distance. The NOE intensity between atoms i and j may be estimated by:

$$\mathrm{NOE}_{ij} = C_{ij} \left\langle r_{ij}^{-6} \right\rangle \tag{5.5}$$

where C_{ij} is a constant for atom types i and j, r_{ij} is the inter-atomic distance and the angle brackets represent the ensemble average. It is difficult to calculate C_{ij} and so the approximation is made that C_{ij} is the same for all atom types. C_{ij} is assigned the value 729 (3^6) so that an inter-atomic distance of 3 Å gives an NOE of unity. If the inter-atomic distance of two protons i and j is less than 3 Å ($\mathrm{NOE}_{ij} \geq 1$) then the signal is said to be "strong". Between 3 and 4 Å ($1 > \mathrm{NOE}_{ij} \geq 0.18$) the signal is said to be of "medium" intensity, and at distances of between 4 and 5 Å ($0.18 > \mathrm{NOE}_{ij} \geq 0.05$) the signal is "weak".

5.6.3: System Set-up

The DL_POLY and DLPROTEIN packages come with some utilities to aid the setting up a simulation system, but they are not sufficiently flexible for this particular case. The aim was to set the system up such that it was as close to the system used by Wu and Wang[60] as possible. Wu and Wang used the CHARMM22 force field[110] and a 29 Å cubic box containing the pentapeptide and 805 TIP3P water molecules.[88] The non-bonded pair-list was cut at 13 Å and a switching function was applied between 8 and 12 Å. The simulation temperature was 300 K, SHAKE was applied to all bonds and a 2 fs time-step was used. DLPROTEIN is capable of applying an electrostatic cut-off but not a residue-based electrostatic cutoff. Using a cut-off not applied on a residue basis will not give a system that conserves energy. The preferred way of calculating the electrostatic interactions in DLPROTEIN is to use the SPME method.[24] However, Ewald methods require a neutral system and so a positive counter-ion had to be placed in the simulation box.

To prepare the DLPROTEIN coordinate input file for solvated YPGDV took many steps. First a Z-matrix of YPGDV was generated by Pepz, part of the MCPRO distribution[7]. The default settings of $-70°$ for the ϕ angle in Pro and 180° for all the other backbone dihedral angles causes a side-chain clash. This clash was removed by adjusting the two dihedral angles between the backbone and the tyrosine ring. A

version of MCPRO, modified to produce cubic solvent boxes, was then used to solvate YPGDV in TIP3P water. The backbone dihedral angles were fixed and MCPRO was used to make 300 000 NVT MC steps—this removed any bad interactions caused by the manner of solvation. The resulting set of coordinates were visualised and a water molecule approximately 16 Å from the peptide was replace with a Na^+ counter-ion. The final system contained 724 water molecules and was converted into a format suitable for DLPROTEIN.

To generate the force field parameters, a PDB file of YPGDV was produced using the xleap program of the AMBER package (ref. 17). The PDB was passed into the dlgen utility of DLPROTEIN which output the parameters for the CHARMM22 force field. Six incorrect entries in the force field were removed and the rest of the force field was checked against the original CHARMM22 parameters. The SPME parameters were chosen according to the method given in the DLPROTEIN manual and the Lennard-Jones non-bonded interactions were cut-off at 12 Å.

Before any data-gathering simulations were performed, an MD simulation of 10 000 steps (20 ps) was performed in the NVT ensemble. All subsequent simulations were restarted from this point.

5.6.4: Filter Design

To help decide what frequencies to target, a 16 678 step simulation in the NVE ensemble was performed. The spectral density of the peptide was calculated but is not presented here as it was no help in the design process. Instead, the amplitude spectra of the eight ϕ and ψ angles were calculated with $1\,\mathrm{cm}^{-1}$ resolution. The cumulative sum of the amplitudes with frequency is shown in figure 5.26. Although the quantitative value of the cumulative sum has no physical meaning, the shape of the graphs show the presence of large-amplitude low frequency motion. The dihedral with the least motion is ϕ_2, because ϕ_2 is part of the proline ring.

To target the dihedral motion while minimising the effect on the other degrees of freedom, a filter designed for amplifying 0–$25\,\mathrm{cm}^{-1}$ and using only 1001 coefficients was created in MATLAB. As with solvated alanine dipeptide, the filter was designed with frequency responses of 1 and 0, and all filters used in this section were derived

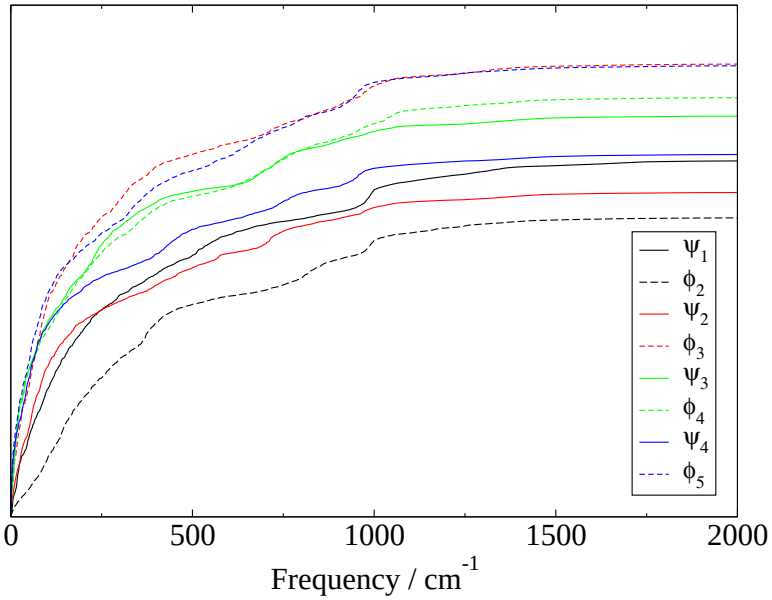

Figure 5.26. The cumulative sums of the amplitude spectra of the eight ϕ and ψ backbone dihedral angles of YPGDV. The quantitative value of the cumulative sum (on the y-axis) has no physical meaning. The graphs show the presence of large-amplitude low frequency motion.

from this one seed filter by multiplying the coefficients through by one less than the required amplification, and then raising the whole frequency response by 1 by adding 1 to the middle coefficient. The filter, shown in figure 5.27 has a very slow transition from the $\mathcal{H}(\omega) = 1$ region to the $\mathcal{H}(\omega) = 0$ region as a result of the relatively small number of coefficients used. By using a small number of coefficients in the design, the amount of data-collection required by the RDFMD steps is minimised, while a filter that generally enhances low-frequency motion is created. The MATLAB command for designing the seed filter is `fircls(1000,[0 0.002998 1],[1 0],[1.1 0.001],[0 -0.001],'text')`. The boundary between the two regions (for a 2 fs time-step) is at 25 cm^{-1}, the low-frequency region is constrained between 0 and 1.1 and the high-frequency region is tightly constrained between -0.001 and 0.001.

To determine the half-life of the energy dissipation in this system (and thus judge the size of the delay required between filter applications) five simulations from five start points were run. In each simulation a 0–25 cm^{-1} $\times 10$ amplifying filter was applied twice using a 15 time-step delay (the 15 step delay was taken from the solvated alanine dipeptide work). The decay of the internal kinetic energy of the pentapeptide was

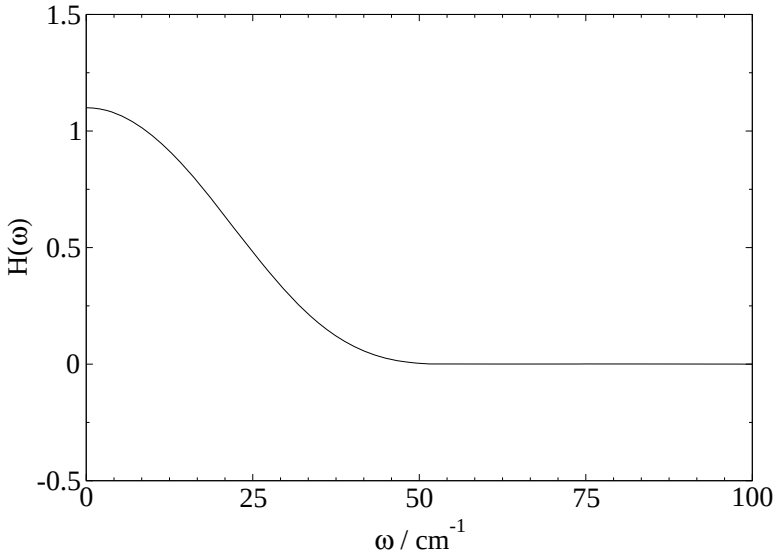

Figure 5.27. The frequency response of the filter designed for YPGDV. All filters used in the pentapeptide work are derived from this filter.

plotted against time and exponential decay curves were fitted to the data. The decay curves fitted to the data resulting from the second filter applications all had correlation coefficients greater than 0.93. The half-lives of these decay curves were 11, 14, 18, 19 and 20 time-steps. These half-lives are similar to those found in solvated alanine dipeptide, and so the same 15 time-step delay was used for the pentapeptide.

5.6.5: Inducing Conformational Change

In the SGMD simulation of YPGDV, the self-guiding force was applied constantly for 2 ns, but it is not possible to apply the RDFMD method constantly. Instead, one hundred 20 ps simulations were performed in the NPT ensemble, alternating with RDFMD filter applications in the NVE ensemble.

Various RDFMD protocols were devised and tested. The protocol must apply the filter until conformational change is found, but it is important not to put too much energy into the system and produce conformational change via high-energy pathways. All the protocols share a common framework:

1. A 1001 step NVE simulation is performed to fill the filter buffer.
2. The filter is applied (to all or a sub-set of the peptide atoms) and two simulations restart from the filter output: one moving backwards in time for 485 steps and one moving forwards in time for 515 steps.

3. A test is applied to determine if the filter application caused conformational change in the forward trajectory. If conformational change is found, then the next NPT simulation restarts from the end of the forward trajectory. If no conformational change is found, then the combined filter buffer of the forward and reverse simulations is used for the next filter application (repeat item 2), moving the system forwards by 15 steps.

A series of conformational change tests were developed. All made use of the clustering algorithm already described (section 5.6.1). The best performing test is described here. The test determines whether the configuration of the peptide at the end of the forward trajectory in step 2 is in a different conformation than any of the configurations in the initial buffer-filling simulation (step 1). During these two simulations, the Cartesian coordinates of the peptide are saved to disc at every step. When the test is applied, the backbone dihedral angles are first extracted from the Cartesian coordinates, and the angles from both the initial trajectory and the forward trajectory being tested are fed into the clustering algorithm. Conformational change is said to have occurred if the cluster containing the final configuration of the forward trajectory is composed of a large proportion (greater then 70%) of configurations from the same forward trajectory,

Both of the RDFMD simulations presented here use the same basic protocol and conformational change test described above. They differ in the manner in which they seek to limit the amount of energy imparted to the system by the filter applications. The first protocol is as above, applying the filter to all the atoms, but limiting the total number of successive filter applications in a single RDFMD series to four. If the filter is applied four times and no conformational change is detected, then the next 20 ps NPT simulation is restarted from the end of the last NPT simulation.

The first protocol produced a surprising conformational transition after the fourth filter application of the very first RDFMD application series. Unfortunately, the transition was the ω backbone dihedral between proline and glycine moving from the usual 180° position to 0°. In general, the 180° position for an ω dihedral angle is very stable and a large input of energy would be required to cause such a conformational change. Analysis of the kinetic energy of the pentapeptide immediately after the forth filter application showed an instantaneous temperature of 1785 K, demonstrating that (in this

case) limiting the number of filter applications to four was not sufficient to prevent excessive energy input.

The second protocol described here is more complicated and is shown schematically in figure 5.28. As before, an initial NPT simulation if performed followed by the RDFMD stage. A 1001 step NVE simulation is run starting from the end of the NPT simulation, and the filter is applied to all of the atoms. NVE ensemble MD then continues with a 515 step forward simulation and a 485 backward simulation. The algorithm then branches with a series of tests. The first test applied is to calculate the internal temperature of the peptide at the first step of the forward simulation (immediately after the filter application). In order to judge the temperature of the internal degrees of freedom, the internal temperature is calculated from the same "internal velocities" as used by the filtering process. If the instantaneous internal temperature exceeds 1500 K, then the RDFMD stage is terminated and the next NPT simulation continues from the end of the previous NPT simulation. (The choice of temperature cut-off was made by examining the internal temperatures in the RDFMD stages of the simulation using the previous protocol.) If the internal temperature is not too high, then the conformational change test is applied. As before, if conformational change is detected then the next NPT simulation is launched, restarting from the end of the 515 step (forward) NVE simulation. If no conformational change is detected and the filter has not been applied ten times then the filter is applied again, otherwise the RDFMD stage is terminated and the NPT simulation restarted. The algorithm was followed until one hundred 20 ps NPT simulations were complete (2 ns NPT in total), during which time no conformational change in any ω angle occurred.

The analysis of the simulation performed using this second protocol is presented below.

5.6.6: Results

In this section, the RDFMD trajectory is compared with the 2 ns SGMD trajectory and with a 2 ns MD simulation run in the NPT ensemble using the same starting point and force field as the RDFMD simulation. In both simulations, the state of the system was saved to disc every 1 ps (twice the rate used by Wu and Wang[60]). The RDFMD

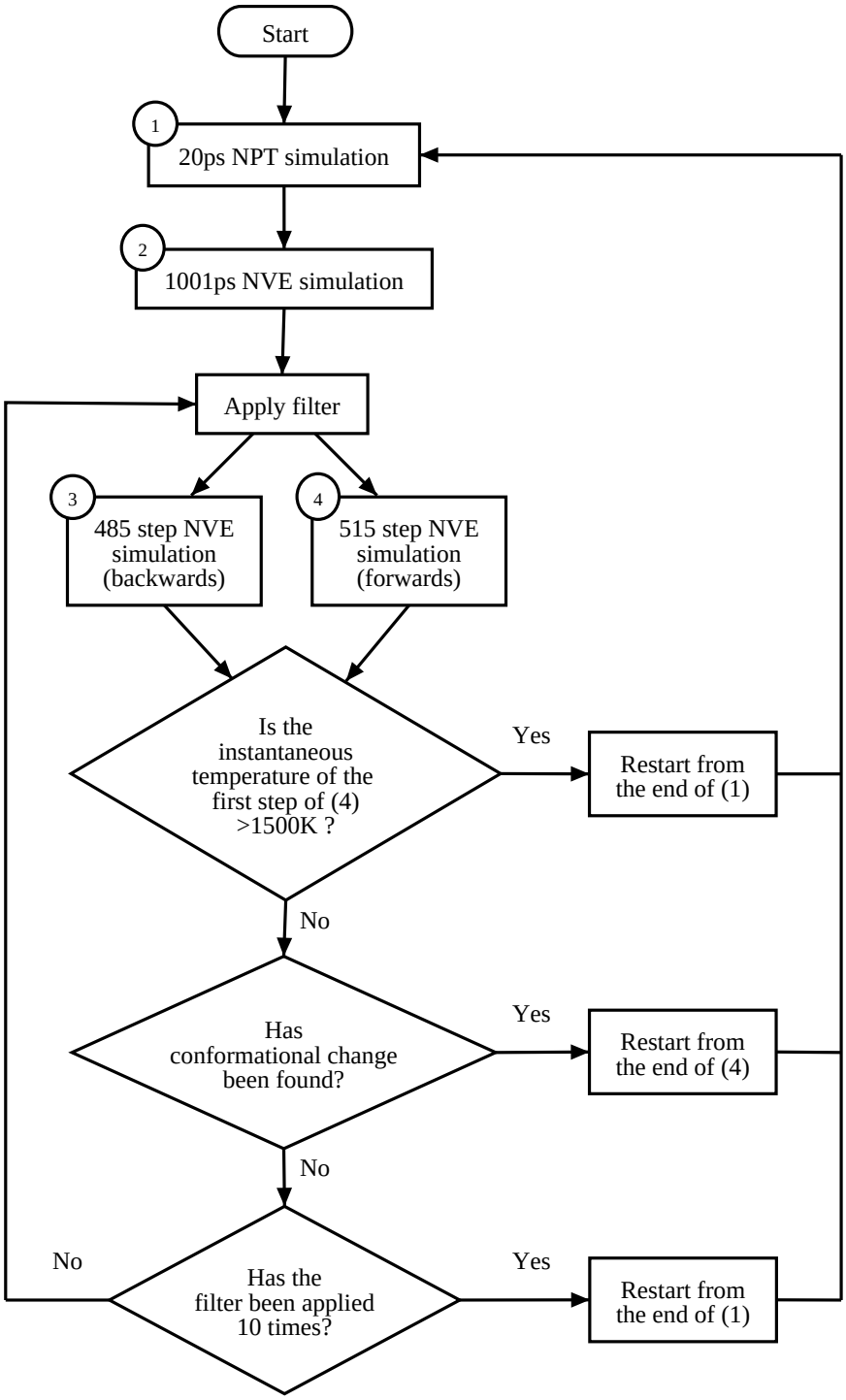

Figure 5.28. One of the protocols used to enhance conformational sampling in YPGDV. A positive response to any of the three questions causes the next 20 ps simulation to be run. The simulation is restarted either from the end of the previous NPT stage or from the final step of the forward NVE simulation (4). As shown, this simulation will never stop, but in the example presented in this thesis, the simulation was terminated after one hundred 20 ps simulations had been performed (2 ns in total).

trajectory was composed of 20 steps for each NPT section plus, at each stage where conformational change was detected and accepted, the two saved steps from the initial buffer-filling 2 ps NVE simulation (simulation 2 in figure 5.28). Fifty-two conformational change events out of one hundred were detected during the filter application stages, and so the total number of points in the RDFMD trajectory numbers 2104.

Figure 5.29 shows the conformational space sampled during the standard MD trajectory. Only the ϕ_3, ψ_3 and ψ_4 dihedral angles are sampled more than a small amount. This is consistent with the chemical structure of YPGDV, in that ϕ_3 and ψ_3 are both in the glycine residue and ψ_4 is in the aspartate residue, and are thus relatively free of steric clash from side chains. In addition, no sampling of ϕ_2 is observed which is consistent with it being constrained by the proline ring.

Figure 5.30 plots the conformational space sampled by the RDFMD trajectory. In comparison with the MD trajectory, the sampling has clearly been increased in both the ψ_2 and ψ_3 dihedrals. However, these plots are of limited use as they only show correlation between pairs of dihedrals—it is not possible to link a point on one of the quadrants with the corresponding point on the other three quadrants. They do not therefore give a complete picture of the secondary structure.

The configurations in both the MD and RDFMD trajectories were clustered according to Wu and Wang's clustering algorithm[60] (see section 5.6.1). Clustering the MD trajectory produced two major clusters accounting for 37% and 15% of the configurations. The six next largest clusters accounted for only a further 12%, and 357 further small clusters were required to include all the configurations. Clustering the RDFMD trajectory produced an unsatisfactory result. The largest cluster was of just 72 configurations (3.4%) and 952 clusters were found in total. A clustering method is supposed to simplify the description of the conformational space sampled—952 clusters from 2104 configurations is not much of a simplification. A larger number of clusters is to be expected in the RDFMD simulation since the filter applications were driven by the target of moving to a new cluster. However, the number of clusters found even in the MD trajectory seems large. The cause of the large number of clusters seems to be insufficient sampling of the trajectory. To investigate the clustering, the second nanosecond of the MD trajectory was configured to save the coordinates to disc every

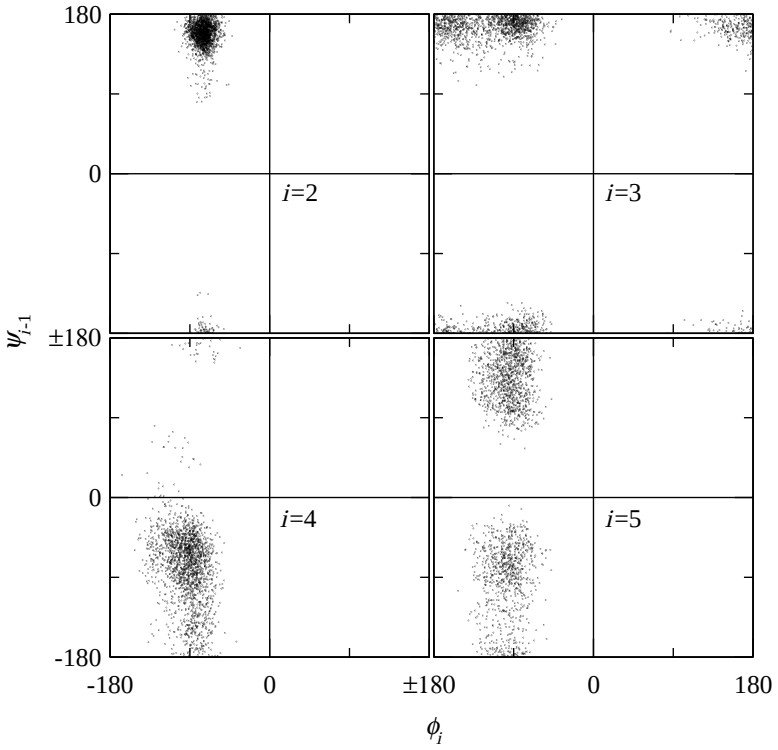

Figure 5.29. Backbone dihedral angle space sampled during the 2 ns MD simulation of YPGDV.

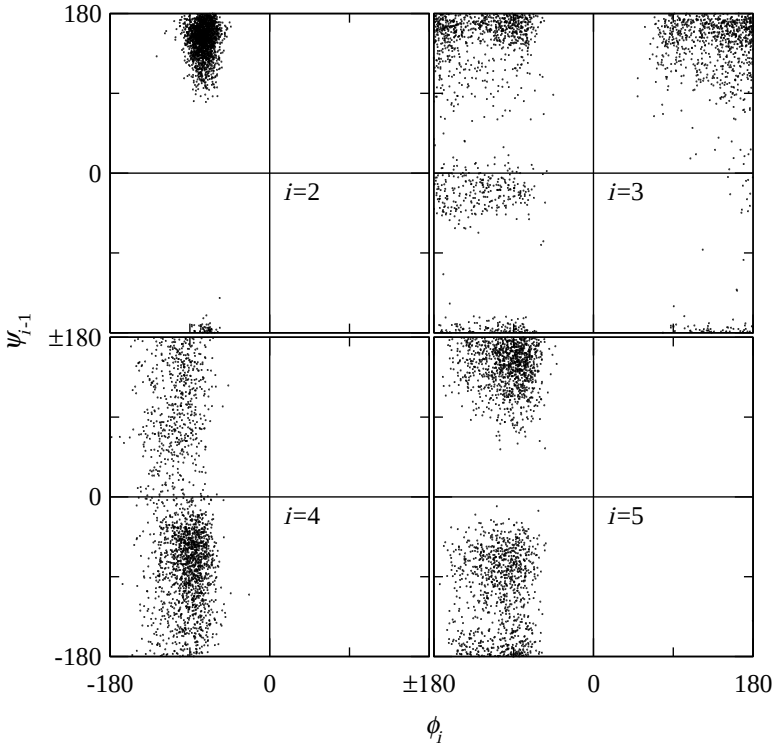

Figure 5.30. Backbone dihedral angle space sampled during the 2.1 ns RDFMD simulation of YPGDV.

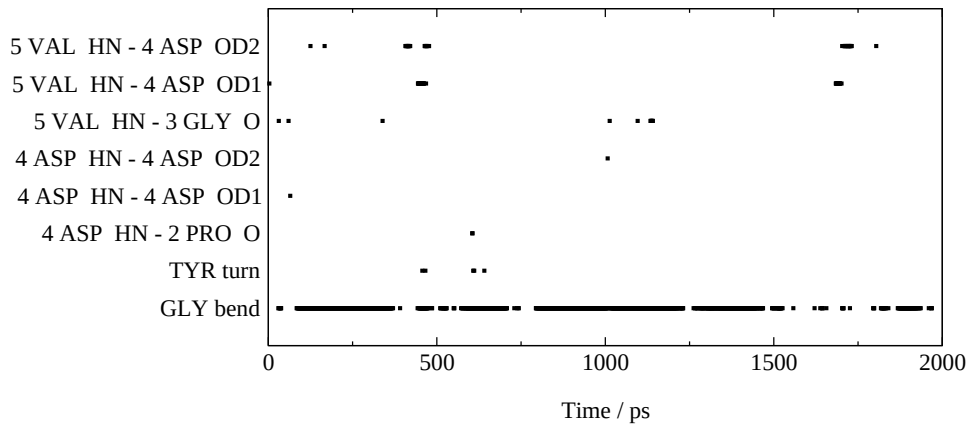

Figure 5.31. The hydrogen bonds and secondary structure elements present in the 2 ns MD simulation.

200 fs. When this part of the trajectory was clustered, the first three clusters accounted for 68% of the configurations and only nine clusters were required to reach 80%. Clustering one configuration in every five however (to give the same sampling rate used in the other simulations) produced clusters where the 72 largest clusters were required to account for 80% of the configurations. Therefore the clustering result is clearly affected by the sampling rate of the simulation. In the RDFMD trajectory, the peptide changes conformation much more rapidly, and so a high sampling rate would be required for this clustering algorithm to work. The clustering algorithm performed very well in the RDFMD filter application stages where the configuration from every time-step was considered but is not useful for clustering complete trajectories with a relatively low sampling rate.

Figures 5.31 and 5.32 show hydrogen bond and secondary structure analyses of the MD and RDFMD trajectories respectively. The analyses were performed by MOLMOL (ref. 111) using the DSSP algorithm[112] for the secondary structure analysis, and the MOLMOL default parameters for the hydrogen bond detection. In the analysis of the MD trajectory, a turn at 1 TYR and a bend at 3 GLY were detected. No other secondary structure elements apart from "coil" were found. By comparison, the RDFMD trajectory found a wider range of hydrogen bonds (including 1 TYR to 4 ASP), more configurations contained hydrogen bonds and more configurations contained the tyrosine turn. Thus, the RDFMD trajectory is more consistent with the NMR results.

As a demonstration of the enhanced rate of conformational change that RDFMD

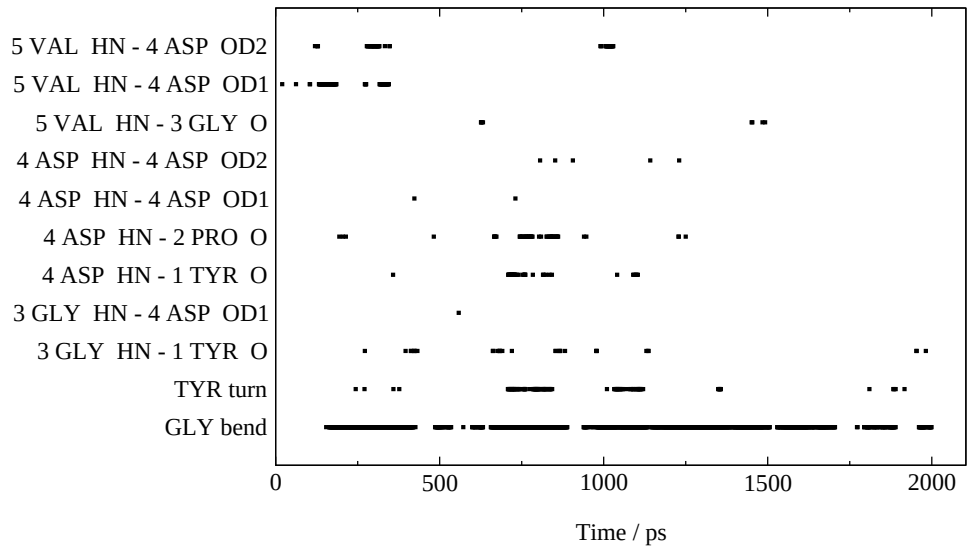

Figure 5.32. The hydrogen bonds and secondary structure elements present in the 2.1 ns RDFMD simulation.

provides, the root mean square deviation (RMSD) of the ϕ and ψ angles of each saved step with the previous saved step was calculated for both the MD and RDFMD trajectories. If θ_i represents one of the eight backbone ϕ and ψ angles, and $\theta_i(t)$ is the value of the angle at time-step t, then the RMSD was calculated by:

$$\mathrm{RMSD}_t = \left(\frac{\sum_{i=1}^{8} [\theta_i(t) - \theta_i(t-1)]^2}{8} \right)^{\frac{1}{2}} \tag{5.6}$$

where t ranges from 2 to 2000. Angle periodicity was taken into account in calculating the difference. Figure 5.33 shows the variation with time of the RMSD for the MD simulation. The mean RMSD is 15.2° and the standard deviation is 5.2°. This is to be compared with the RDFMD data shown in figure 5.34. The mean and standard deviation of the RMSD for the RDFMD simulation are 18.0° and 7.5° respectively. The mean and standard deviation of the RMSD's immediately following a filter application (marked by the red lines in figure 5.34) are 30.2° and 13.7. This demonstrates a large degree of conformational change following a filter application.

The NOE intensities for the MD and RDFMD simulations were calculated and assigned according to the algorithm presented in section 5.6.2. They are listed, along with the SGMD values and experimental assignments in table 5.6. The experimental assignments tabulated in Wu and Wang's paper[60] (of "very strong", "moderately strong", "strong", "weak" and "medium" respectively) are not the same as those in the original

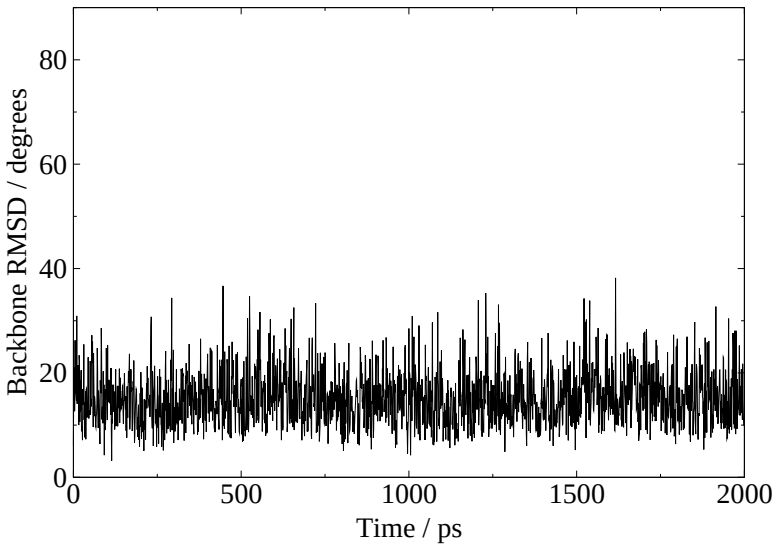

Figure 5.33. The RMSD of the ϕ and ψ angles at each step with the previous step for the MD simulation of YPGDV.

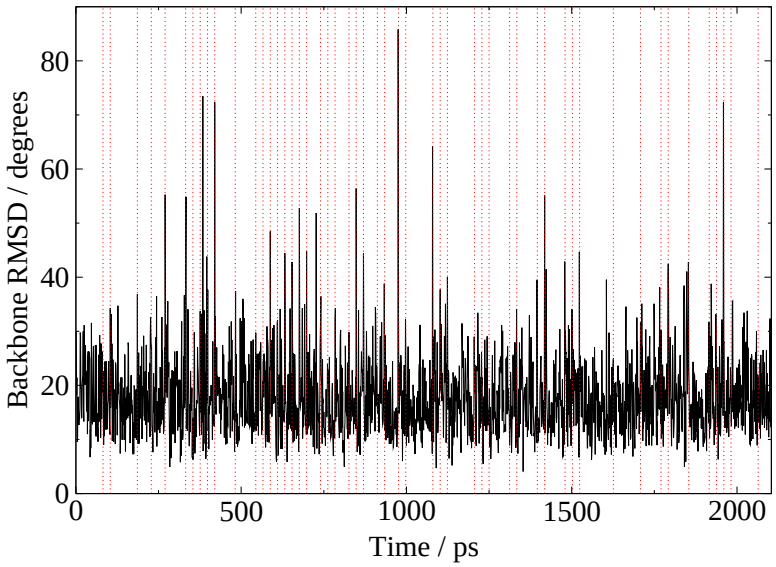

Figure 5.34. The RMSD of the ϕ and ψ angles at each step with the previous step for the MD simulation of YPGDV. The red lines mark the steps immediately following filter applications where conformational change was detected.

paper by Dyson *et al.*[107]. Also, a strong peak intensity for $d_{\alpha N}(4,5)$ appears in Wu and Wang's paper but is not tabulated in the Dyson paper (it is present in the figure of the original spectrum however). A personal communication with both H. Jane Dyson and Peter E. Wright (two of the authors of the experimental paper) resulted in three conclusions:

1. Wu and Wang have over-interpreted the original experimental data. Assignments

of anything more precise than "weak", "medium" and "strong" are not advisable.

2. The $d_{\alpha N}(4,5)$ peak is indeed strong. It was not tabulated because it provides no insight into the structure of the turn. This peak is strong in any unstructured protein.
3. The original assignment of "strong" for the $d_{NN}(3,4)$ peak "probably reflected the fact that d_{NN} NOEs are usually weak and frequently unobservable in peptides but this one is unusually strong." It is definitely weaker than the $d_{\alpha N}(2,3)$ peak though.

Table 5.6. The experimental and calculated NOE intensities and assignments for YPGDV. An NOE $\geq$ 1 is assigned "strong", 1 $>$ NOE $\geq$ 0.18 is "medium" and 0.18 $>$ NOE $\geq$ 0.05 is "weak".

Connectivity	Experiment	MD	RDFMD	SGMD
$d_{\alpha N}(2,3)$	s	3.733 (s)	4.024 (s)	8.540 (s)*
$d_{\alpha N}(3,4)$	m	0.567 (m)	1.768 (s)	2.240 (s)
$d_{\alpha N}(3,4)^\dagger$	m	3.940 (s)	3.170 (s)	
$d_{\alpha N}(4,5)^\ddagger$	s	3.685 (s)	3.226 (s)	3.380 (s)
$d_{\alpha N}(2,4)$	w	0.071 (w)	0.084 (w)	0.240 (m)
$d_{NN}(3,4)$	s	2.599 (s)	2.478 (s)	0.880 (m)

$\star$ This intensity was assigned "very strong" in the original SGMD paper (see main text).
† The original spectrum has two medium intensity peaks for this resonance arising from the two C_α protons on 3 GLY. The SGMD paper quotes just one computed value.
‡ The $d_{\alpha N}(4,5)$ peak is not listed in the original paper by Dyson *et al.* because it provides no insight into the structure of the turn. This NOE is strong in any unstructured peptide.

Care must be taken not to over-interpret the NOE results. One aspect that stands out is the calculation of the two $d_{\alpha N}(3,4)$ peaks. These resonances show up as almost identical separate peaks in the original spectrum, both of medium intensity. In the MD simulation, the calculated NOEs are so different that one falls into the "medium" band and one into the "strong" band. In the RDFMD simulation the calculated NOEs are much more similar, suggesting more even sampling of ψ_3. Looking back at the ϕ-ψ plots (figures 5.29 and 5.30), this is indeed the case. Only one value for $d_{\alpha N}(3,4)$ was presented in the SGMD paper so no comparison of NOEs can be made. However, the ϕ-ψ plots in the SGMD paper show better sampling of the ψ_3 degree of freedom in their normal MD simulation than the SGMD simulation.

The rankings in order of NOE intensity for the different connectivities in the RDFMD and SGMD simulations are the same (i.e. both find $d_{\alpha N}(2,4)$ to be the weak-

est, then $d_{\mathrm{NN}}(3,4)$, etc). Also, apart from the improvement in the $d_{\alpha N}(3,4)$ pair, the MD and RDFMD results are very similar. This is despite the dramatically difference in ϕ-ψ space sampled, and different secondary structure plots for the two methods. The imprecision of the experimental NOE intensities means that all that can be said is that all three methods (MD, RDFMD and SGMD) are in reasonably good agreement with experiment.

5.6.7: Summary

The conformational clustering technique and the method of calculating NOE intensities used by Wu and Wang[60] have been described and applied to the analysis of a 2 ns MD simulation and a 2.1 ns RDFMD simulation of YPGDV solvated in water. The system was set up to mimic as closely as possible the system used by Wu and Wang for their SGMD simulations. An initial simulation was performed to determine what design of filter was required for amplifying conformational change, and a study of the energy dissipation half-life was made to determine the delay between filter applications. These studies resulted in a 0–25 cm^{-1} amplifying filter with a 15 step delay. Two protocols for inducing conformational change were designed. The simplest protocol failed in that it over-heated the dihedral modes and caused a conformational transition in an omega dihedral angle. The more complicated protocol employed an internal temperature cut-off to prevent this. A 2.1 ns RDFMD simulation using the advanced protocol was performed and compared to the MD and SGMD trajectories (where possible). It was found that the configurations generated during the RDFMD experiment were better than the MD configurations in terms of secondary structure elements, that conformational change events often occurred soon after the filter application, and that the NOEs calculated from the RDFMD trajectory were in closer agreement to the experimental data than those of the MD trajectory and, arguably, the SGMD trajectory.

5.7: Conclusion

The limitations of the DFMD method, listed at the start of the chapter, are the difficulty in targeting low-frequency oscillations in systems where energy dissipation is rapid (e.g. condensed phase systems), and the requirement to introduce all the energy into the

system in a single multiple-pulse filter application. The RDFMD method improves on the DFMD method by applying the filter to trajectories sampled both forwards and backwards in time. This enables the user to choose exactly where in the trajectory the filter is applied. By choosing to space filter applications closely in the trajectory, one filter application may build on the energy input and enhanced trajectory of a previous application in a way that is not possible with DFMD. The RDFMD method thus provides a mechanism for introducing frequency-targeted energy into the system in a gradual manner.

To decrease the overhead associated with filling the filter buffer, a study was made into filter design methods. The best filter design method found in the study was the *fircls* function of MATLAB. The function permits the user to specify regions of the frequency response where the exact response is unimportant. By relaxing the constraints in these regions, the other regions can be more precisely constrained. It was found that the *fircls* function could produce filters of equivalent quality to the original Fourier method using approximately 70% of the coefficients.

The RDFMD method has been applied to single molecules of pentane and alanine dipeptide, solvated alanine dipeptide and solvated YPGDV. The study of pentane provided evidence that introducing the energy gradually could promote more reasonable transitions than introducing all the energy in one step. The alanine dipeptide simulations (for which the adiabatic potential energy surface was calculated) showed a change in the characteristic frequency of the target mode as energy was gradually introduced into the system. The change in the frequency was accounted for by using a succession of different filters. It was also determined that the ϕ and ψ angles in alanine dipeptide oscillated with different frequencies, and the frequency-specific nature of the RDFMD method was demonstrated by causing transitions driven by either ϕ or ψ. Reasonable conformational transitions were successfully induced by the RDFMD method in both these gas-phase systems.

Successful conformational change events via a variety of different transition paths were induced in alanine dipeptide solvated in chloroform from fourteen out of fifteen different start points. The energy introduced into the system for each conformational change was reproduced using a heating pulse. No conformational transitions were pro-

duced by adding the same amount of energy into the system in this non frequency-specific manner.

Finally, the RDFMD method was applied to the pentapeptide solvated in water. A complex protocol for performing an NPT simulation with regular RDFMD filter applications was devised. The resulting trajectory was compared with a normal MD trajectory and with limited data on the SGMD trajectory for the same system. The configurations generated during the RDFMD experiment had more defined secondary structure than those of the MD trajectory and gave better agreement with the experimental NOE data than either the MD or SGMD trajectories.

6: Spectral Analysis Methods

In this chapter, Fourier methods of spectral analysis (including the spectral density) and their associated problems will be discussed. The Hilbert transform and the Empirical Mode Decomposition (EMD) algorithm will then be introduced. Finally, the Hilbert-Huang Transform (HHT) will be discussed and applied to the analysis of MD trajectories.

6.1: Fourier Methods

The Fourier methods of spectral analysis include the basic Fourier transform (used in this thesis for calculating spectral densities), the spectrogram and wavelet analysis. The Fourier transform and its associated problems have already been discussed in chapter 3. To summarise: the Fourier transform takes a data-set, replicates it to infinity both forwards and backwards in time, and calculates the linear combination of sine waves necessary to reproduce the replicated data-set. The Fourier transform is a mathematically exact method but its physical interpretation is difficult. Discontinuities in the infinite data set (arising either from the original data or from the replication process) cause leakage away from the true frequencies. The same problem occurs if the data-set contains time-localised frequencies. The frequency leakage is an unavoidable aspect of the

Fourier transform because it must fit invariant sine waves to a time-localised signal. A non-periodic signal, or one with time-localised events, is said to be a "non-stationary" data-set. Molecular dynamics simulations in the condensed phase inevitable produce non-stationary data.

The Fourier transform is the most commonly used method of spectral analysis, so much so that the term "spectrum" is often taken to mean the Fourier spectrum. However, great care must be taken in the physical interpretation of Fourier spectra (and derivatives such as the spectral density) and uncritical use must be avoided.

A modification of the basic Fourier technique is the spectrogram.[113] To calculate a spectrogram, the Fourier transform is applied using a window that cuts out all the data apart from a localised time section. By sliding the window along the time axis and repeatedly calculating the Fourier transform, a time-frequency distribution can be obtained. For instance, the Fourier transform of the first 5 ps of a simulation may be compared with the Fourier transform of the final 5 ps of data. Since the spectrogram method relies on the Fourier transform, the same problem of non-stationarity in the data occurs. With the spectrogram method the data in each time-window must be stationary, or the same spectral leakage effects of the basic Fourier transform will return. Even if the data is stationary within a time window, the spectrogram method has an additional problem of limited frequency resolution. To obtain precise time information a narrow time-window must be employed. The frequency resolution is inversely proportional to the length of data analysed (see equation 3.12), so the Fourier transform of a small time window gives a low frequency resolution. One cannot obtain both time-localised and frequency-localised information.

Wavelet analysis is a generalised spectrogram method. Whereas the spectrogram essentially fits sine waves to portions of the data-set, the wavelet method fits a function chosen by the user. Any function may be used, but most functions will not give a physically meaningful result. Commonly, the Morlet wavelet function is used.[114] The Morlet wavelet is a complex function defined as the product of a Gaussian and a cosine wave for the real component and the product of the same Gaussian and a sine wave for the imaginary component. The Gaussian is chosen to tail the oscillations off after 5.5 wavelengths to provide time-localised information. To obtain the wavelet spectrum, the

product of the data-set with the wavelet at all possible scales and times is taken. For instance, the wavelet is scaled to be the same length as the data-set and its product with the data-set is calculated. This single point is the datum corresponding to the lowest detectable frequency. The wavelet is then scaled to fit one less data point than the length of the data-set, and the product of the wavelet with the data-set is taken at two positions—one where the start of the wavelet lines up with the start of the data-set and one where the ends of the wavelet and the data-set are aligned. The wavelet is then shrunk by another data point and three products are calculated. This process is repeated until the Nyquist limit is reached. The frequency information comes indirectly from the scale of the wavelet applied. The product of the wavelet with the data-set provides a measure of intensity and the position of the wavelet in time provides the time location.

The use of wavelets in the analysis of MD trajectories is limited by its poor time definition at low frequency. However, of the three techniques discussed, the wavelet transform is the best method for analysing non-stationary data and has become extremely popular in the field of image analysis and compression.[115]

6.2: Hilbert Transform

Whereas the Fourier transform takes a time-domain signal, $x(t)$, and moves it into the frequency domain, the Hilbert transform[116] of $x(t)$ produces another time-domain signal. Specifically, the Hilbert transform of a real-valued function $x(t)$ extending from $-\infty < t < \infty$ is another real-valued function $\tilde{x}(t)$ defined by:

$$\tilde{x}(t) = \int_{-\infty}^{\infty} \frac{x(u)}{\pi(t-u)}\, du \tag{6.1}$$

This is the convolution of $x(t)$ with $1/\pi t$.

Consider calculating the Hilbert transform of a simple cosine wave. As t in equation 6.1 is varied, the $1/\pi(t-u)$ function is translated along the cosine wave, causing its asymptotic centre to be placed at t. At time $t = 0$ the asymptotic function is lined up with a maximum in the cosine wave, this is illustrated in figure 6.1. $\tilde{x}(0)$ is the integral of the product of the two functions. In this case the integral is zero because

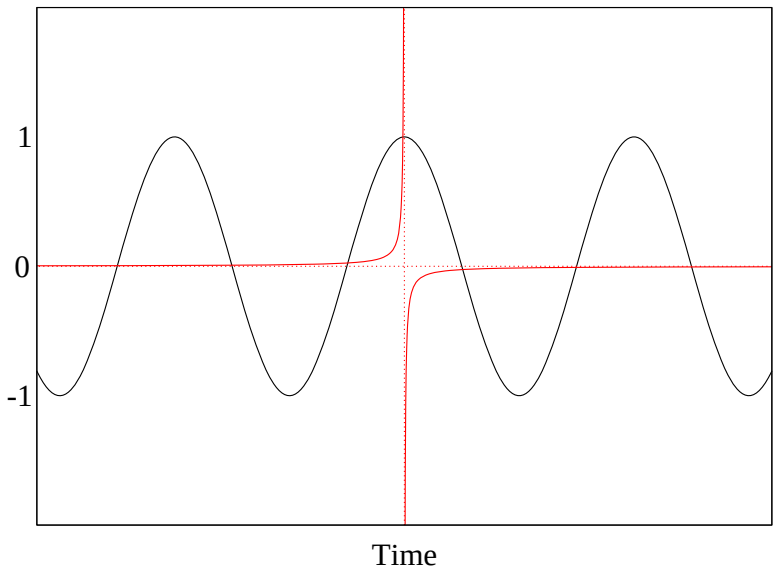

Figure 6.1. To calculate the Hilbert transform of a cosine wave at time zero, the product of the cosine wave (black) and the $1/\pi t$ function (red) is integrated.

the product is of an even function ($\cos(u)$) with an odd function ($1/-u$). At another value of t, the asymptotic function will be lined up with a crossing point in the cosine wave. The product will then be the product of two odd functions and its magnitude will be maximised. Between these extremes, the Hilbert transform is continuous and well-behaved. The Hilbert transform of a cosine wave is simply a sine wave of the same frequency and amplitude.

More generally, it may be shown[116] that the Hilbert transform of a signal, $x(t)$, leaves the magnitude of $x(t)$ unchanged but changes the phase by $\pi/2$ (though no calculation of the phase is performed). The rapid diminution of the $1/(t-u)$ function means that its product with the signal is heavily biased to points close to t. The Hilbert transform thus acts on time-localised data.

The original signal, $x(t)$, and its Hilbert transform, $\tilde{x}(t)$, may be considered to be part of a complex signal $z(t)$:

$$z(t) = x(t) + i\tilde{x}(t) \tag{6.2}$$

This can also be written in terms of amplitude, A, and phase, ϕ:

$$z(t) = A(t)\mathrm{e}^{i\phi(t)} \tag{6.3}$$

where

$$A(t) = \sqrt{x^2(t) + \tilde{x}^2(t)}$$
$$\phi(t) = \tan^{-1}\left(\frac{\tilde{x}(t)}{x(t)}\right) \tag{6.4}$$

Knowing the phase of the signal at time t is useful because we can then consider the rate of change of the phase angle with time. The phase angle of a high frequency motion will change more quickly than the phase angle of a lower frequency motion. The relationship between frequency and rate of change of phase is linear and the instantaneous frequency, $f(t)$, is defined as:

$$f(t) = \left(\frac{1}{2\pi}\right)\frac{d\phi(t)}{dt} \tag{6.5}$$

To summarise, if we have a signal, $x(t)$, we can calculate the signal's Hilbert transform, $\tilde{x}(t)$, and by combining $x(t)$ with $\tilde{x}(t)$ we can obtain the signal's phase. The derivative of the phase with respect to time gives the signal's instantaneous frequency, $f(t)$. The instantaneous frequency of a sine wave would be constant, but for more complicated waves the frequency could vary with time. The method has potential for investigating the changes in the frequencies of motion occurring during MD simulations. Unfortunately, to obtain a meaningful and well-behaved instantaneous frequencies, the wave to be analysed must be like a sine wave that varies slowly in frequency and amplitude. This type of wave is not common in real world situations and so the Hilbert transform has, until recently, only rarely been applied to real-world data.

In practice, the Hilbert transform may be computed by taking the Fourier transform of the data, setting to zero all the negative-frequency components, doubling the positive-frequency components and back-transforming the data.[116] This method has been used by Huang *et al.*[117] but causes some problems. The Fourier transform replicates the data-set to infinity in time both forwards and backwards and transforms the infinite data set. By using the Fourier transform to compute the Hilbert transform, the Hilbert transform is also working on this infinite data-set. This can cause ripples in the Hilbert spectrum at the ends of the (un-replicated) data-set because of the discontinuity in the data produced when it is replicated. Other methods for calculating the Hilbert transform are available, but for computational computational ease and efficiency the

Fourier method has been used in this work. The artifacts introduced by this method were not found to be significant.

6.3: Empirical Mode Decomposition

Empirical Mode Decomposition[117] (EMD) is a new method of signal analysis, not published until a year after the work presented in this thesis was started. The aim of EMD is to decompose a signal (which may be non-stationary) into a set of "intrinsic mode functions" (IMFs), where the characteristics of each IMF are such that they may be Hilbert transformed. Thus, through the Hilbert transform, the instantaneous frequency (or frequencies) of the input signal at any point in time may be calculated.

An IMF is defined as a wave in which: (a) the number of extrema and the number of zero-crossings differ by at most one; and (b) at any point, the mean of the envelope defined by the local maxima and the envelope defined by the local minima is zero. This definition satisfies the conditions necessary for the Hilbert transform to work and also suggests the algorithm to use for decomposing the input data into these IMFs. Figures 6.2 and 6.3 show the EMD algorithm. The algorithm proceeds by subtracting each recovered IMF from the original data-set until either the recovered IMF or the residual data is too small in the sense of the integral of its absolute value. The sum of the IMFs and the residual data is the original signal. To find an IMF, the local mean is repeatedly subtracted from the data-set until the number of extrema and zero-crossings in the residual data differ by at most one (this process is termed "sifting"). The local mean is found by taking the mean of a curve through all the maxima and a curve through all the minima.

Figure 6.4 illustrates how the EMD algorithm obtains the first IMF for one coordinate of the Lorenz attractor. Ten sifting steps were needed to obtain the IMF in this case (three intermediate stages are shown in the figure).

6.4: Hilbert-Huang Transform

The combination of the EMD method with the Hilbert transform provides a potentially powerful analysis tool, recently named the Hilbert-Huang Transform (HHT). The method had been applied in various fields, including those of biology,[117, 118] geo-

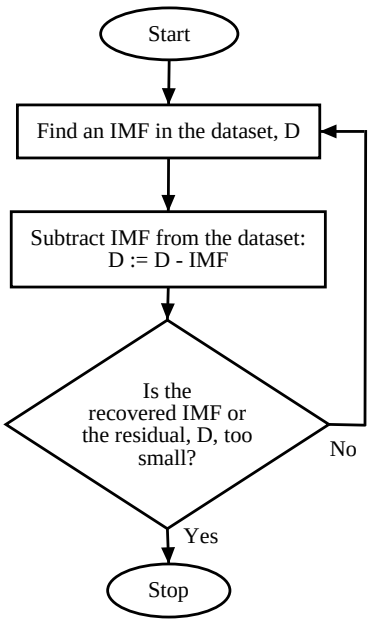

Figure 6.2. The top-level algorithm for finding all the IMFs in a data-set. The algorithm to "Find an IMF" is presented in figure 6.3.

physics,[119] and solar physics.[120] In this section, the method will be tested for some simple signals and then applied to data taken from MD simulations already discussed in this thesis. The programs used to perform both the EMD algorithm and the Hilbert transform were written by Mr. Robert J. Gledhill.

Once a data-set has been decomposed into its IMFs, each IMF is separately Hilbert transformed. The output of the Hilbert transform is the instantaneous frequency and amplitude (or energy) at every point in time along the IMF. The data for all the IMFs is commonly plotted on a single histogram chart with time on the x-axis, frequency on the y-axis and amplitude (or energy) represented by a colour scale. In operation, the frequency axis is split into a number of histogram bins (chosen by the user) and the energy at each time for each IMF is accumulated in the frequency bins. Figure 6.5 shows an example data-set and its Hilbert transform. The data-set is already an IMF and so does not need passing through the EMD algorithm. The Hilbert transform clearly shows the abrupt change in frequency and picks out the two frequencies with precision. The high-frequency wave is correctly coloured as higher energy than the lower frequency wave (see equations A.8 and A.11).

The time-axis of the Hilbert transform may be integrated across to give a frequency

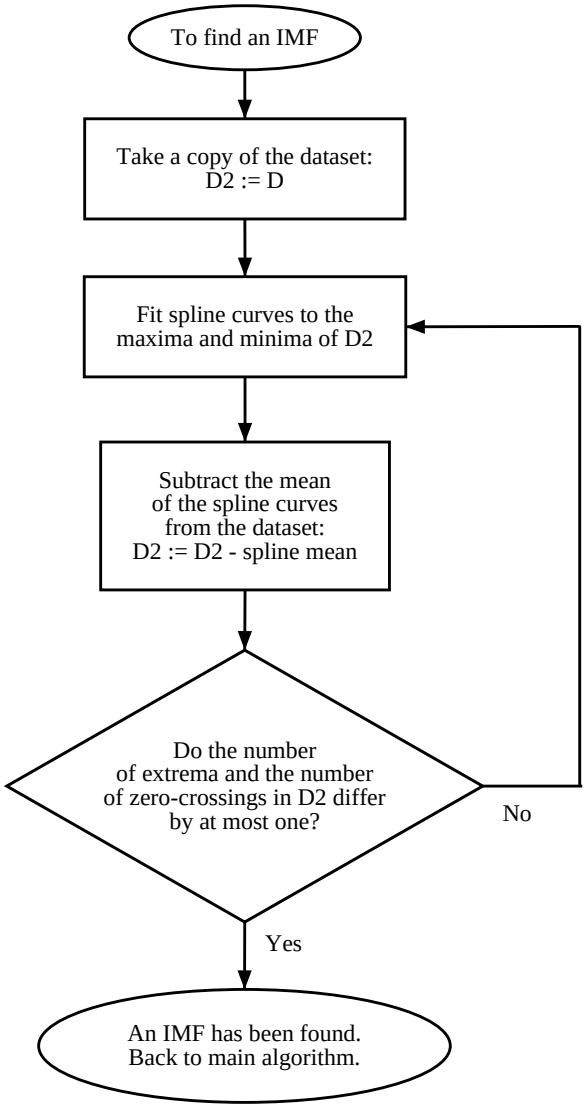

Figure 6.3. The algorithm for finding a single IMF.

versus energy plot similar to the spectral density. Figure 6.6 compares the spectral density of the data-set with the Hilbert frequency/energy spectrum where the frequency resolution of the Hilbert spectrum has been chosen to match that of the spectral density. The two plots are very similar but the Hilbert spectrum has sharper peaks and no ripples. It is only in very simple cases such as this that the Hilbert spectrum can sensibly be compared to a spectral density. As has been noted earlier, if the data-set is non-stationary or non-linear then much of the data in the spectral density is an artifact of the Fourier transform and has no physical meaning, whereas the existence of a frequency in the time-integral of the Hilbert spectrum means that that frequency has

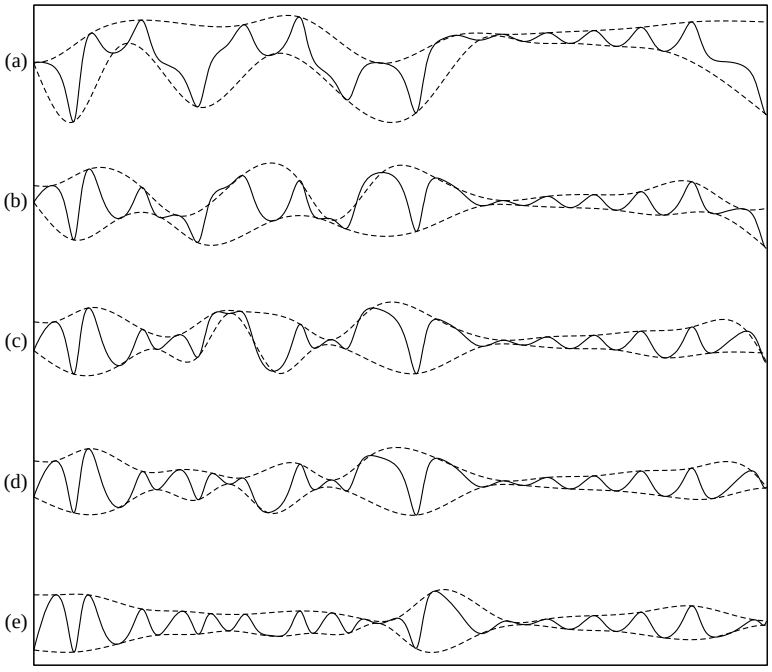

Figure 6.4. Demonstration of the EMD algorithm for one coordinate of the Lorenz attractor. Each part, (a)–(e), shows a stage in the sifting process used for obtaining the first IMF. Solid lines represent the data-set and dashed lines plot the spline curves fitted to the maxima and minima. Between each stage, the mean of the two spline curves is subtracted from the data-set. Part (a) shows the original data and parts (b), (c) and (d) are successive stages. Part (e) shows the final IMF in which the number of extrema and the number of zero-crossings in the data differ by at most one.

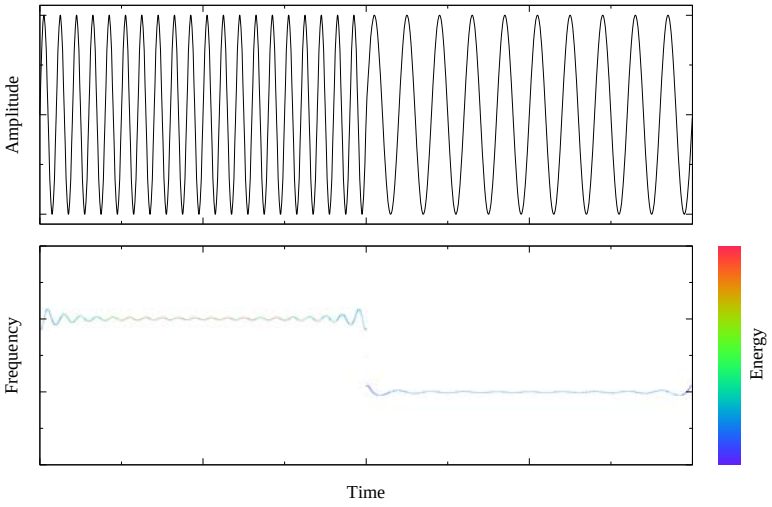

Figure 6.5. A simple data-set with an abrupt change of frequency (top) and its Hilbert transform plotted with an energy colour scale (bottom).

existed locally at some point in time.

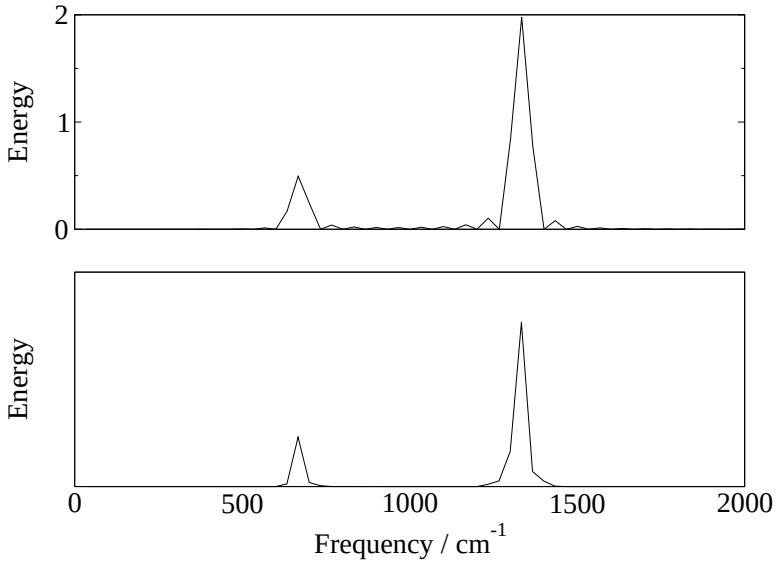

Figure 6.6. The spectral density (top) and the Hilbert frequency/energy spectrum with a linear energy scale (bottom) of the wave shown in figure 6.5.

To further investigate the properties of the EMD/Hilbert method, a test signal was constructed composed of the sum of two sine waves. The first sine wave had frequency $100\,\mathrm{cm}^{-1}$ and amplitude 1, and the second wave had frequency $300\,\mathrm{cm}^{-1}$ and an amplitude varying linearly from 0.2 to 5. The EMD separation of these two waves is almost perfect and the Hilbert spectrum of both IMFs is shown in figure 6.7. The definition of the frequencies (although not displayed in the figure) was excellent and the trend in the energies was correct. A perfect spectrum would show the low frequency oscillation in a single colour throughout the time period. In this case, the EMD algorithm has not produced the original sine waves from the combined signal, rather the low-frequency IMF increases in amplitude slightly with time. The increase in energy of the high frequency wave is not found to be smooth partly because of the low sampling rate of the data. Points are missing from both ends of the spectrum of this (short) data-set because it is not possible to differentiate the phase of the composite Hilbert signal to get the instantaneous frequency at the extremes of the data-set.

In the final test data-set, the sum of two sine waves was once again taken. On this occasion, the amplitudes of both waves were fixed at 10 and the frequency of one wave was set to $100\,\mathrm{cm}^{-1}$. The frequency of the other wave was varied linearly from $25\,\mathrm{cm}^{-1}$ to $666\,\mathrm{cm}^{-1}$ but the sampling rate was chosen to give a maximum sampled

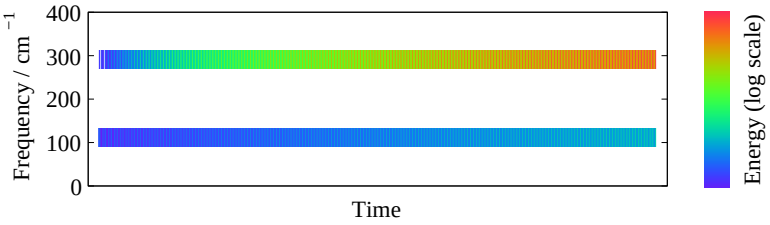

Figure 6.7. The Hilbert spectrum of two fixed-frequency sine waves. The low frequency wave has an amplitude of 1 and the amplitude of the high frequency wave varies linearly from 0.2 to 5. The number of frequency histogram bins was set to 11 and the energy was plotted on a logarithmic scale to give a clear picture of the change in energy in each wave.

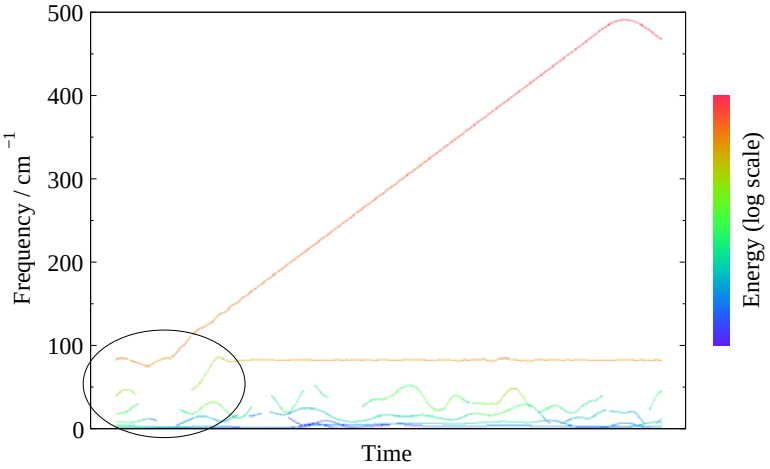

Figure 6.8. The Hilbert spectrum of two fixed-amplitude sine waves. One wave has a frequency of $100\,\mathrm{cm}^{-1}$ while the frequency of the other sine wave varies linearly from $25\,\mathrm{cm}^{-1}$ to $400\,\mathrm{cm}^{-1}$. The energy is plotted on a logarithmic scale. The region where the EMD algorithm has failed is ringed.

frequency of $500\,\mathrm{cm}^{-1}$. Figure 6.8 plots the Hilbert spectrum of this wave. There is clearly a problem with the spectrum during the time period when the ratio of the two frequencies is approximately less than two (this region is ringed in the figure). The physical interpretation of the very low frequency (and low energy) data points is also unclear. For the majority of the data-set though, the two clearest signals are very sharply defined to the correct values. Also, the rebounding of the high frequency signal when it reaches approximately $500\,\mathrm{cm}^{-1}$ (the Nyquist limit) reminds us of the necessity with any analysis method to sample the data sufficiently.

The problems in figure 6.8 are caused by the EMD method, not the Hilbert transform. A similar problem has also been found by Huang[117] and attributed to inaccuracies

introduced by the spline fitting procedure. Even though the HHT method seems to have problems in some circumstances, it will be demonstrated below that useful and consistent data can be extracted from MD trajectories using the procedure.

6.4.1: Application to Molecular Dynamics

In this section, the HHT transform is applied to data-sets obtained from several of the MD simulations already discussed in this thesis. The first simulation to be re-analysed is the conformational transition in gas-phase pentane (section 5.3) induced by the RDFMD method using a 0–150 cm^{-1} amplifying filter to add energy to the low frequency modes very gradually. The trajectory of the pentane molecule is plotted in figure 5.9 part (b) (page 85). For this analysis, the simulation was extended both forwards and backwards in time. This is not necessary for analysis purposes, but was done to provide a clearer picture of the spectrum during the time when no conformational transition is taking place.

Figure 6.9 shows the trajectories of the two dihedral angles in pentane (labelled θ_1 and θ_2), the HHT plot of the dihedral angles, and the HHT plot of just the frequency range amplified by the filter. Not all the points in each IMF are displayed. The mean energy of each IMF is calculated and all points with energy less than half the mean are not displayed. This is to help pick out the important high-energy regions. The filter is applied at the centre of the trajectory (10 ps) and the transition occurs shortly afterwards. The middle part of figure 6.9 shows that while the pentane molecule is in the alternate conformation the dihedrals oscillate at a higher frequency. The bottom part of the figure shows an increase in the low frequency energy as each transition occurs.

Figure 6.10 shows the HHT plot for just the θ_1 angle. From this plot it is clear that very low frequency energy (at approximately 10 cm^{-1}) appears in the system at 10 ps and gradually dissipates, reaching half the mean value of that mode after approximately 5 ps. During the transition into and out of the alternate conformation, the amount of energy in the 25 cm^{-1} to 50 cm^{-1} region increases.

As has been noted earlier, the HHT spectrum can be integrated in time to produce a spectral density equivalent. This is not useful when the data-set is non-stationary as in this case. What is useful is to integrate the HHT spectrum in the frequency axis

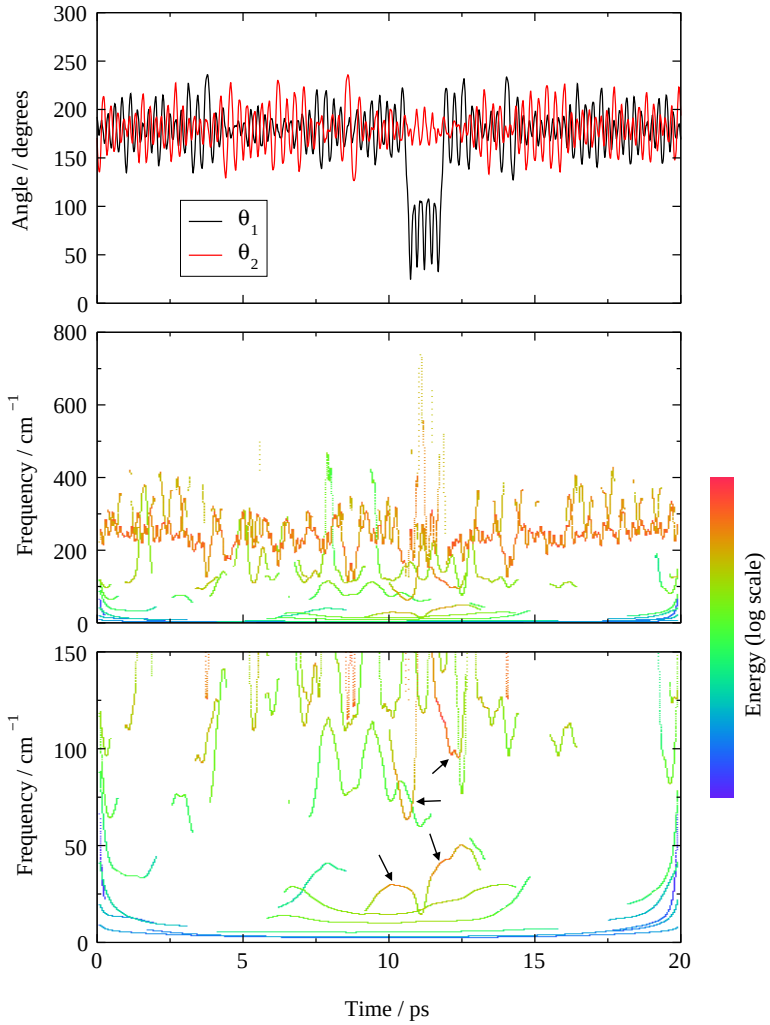

Figure 6.9. Top: the two dihedral angles of pentane. Middle: the HHT plot of the dihedral angles. Bottom: the low frequency region of the HHT plot. Both HHT plots use a logarithmic energy scale and use an energy cut-off of half the mean. The regions where the low-frequency IMFs increase in energy during the transitions are marked by arrows.

to provide a measure of the variation with time of the amount of energy in any chosen frequency band. The frequency integration from 0–55 cm^{-1} of figure 6.10 is shown in figure 6.11. The integration captures just the lowest frequency IMFs and clearly shows the large increase in low-frequency energy during both transitions.

The second simulation to be re-analysed is the gas-phase alanine dipeptide system (see section 5.4). The first analysis presented here is of the simulations performed to investigate the manner in which the frequency of the target mode shifted to a lower value when energy was introduced. To recap: the target mode was identified as 50 cm^{-1}

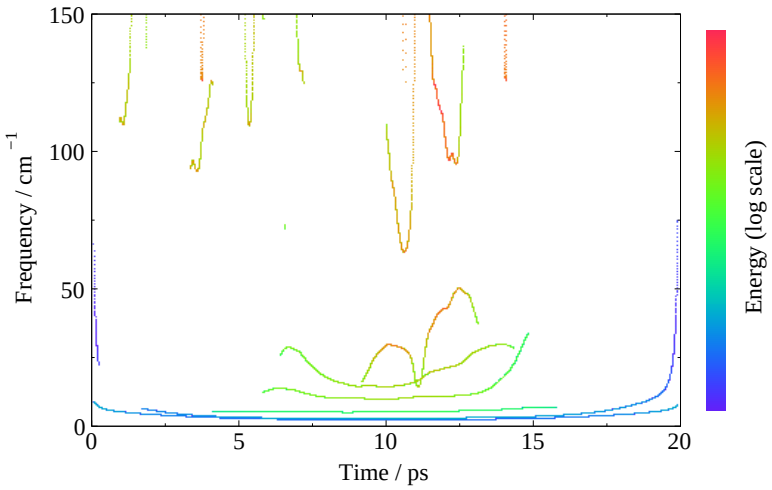

Figure 6.10. HHT plot of pentane dihedral θ_1. An energy cut-off at half the mean has been applied.

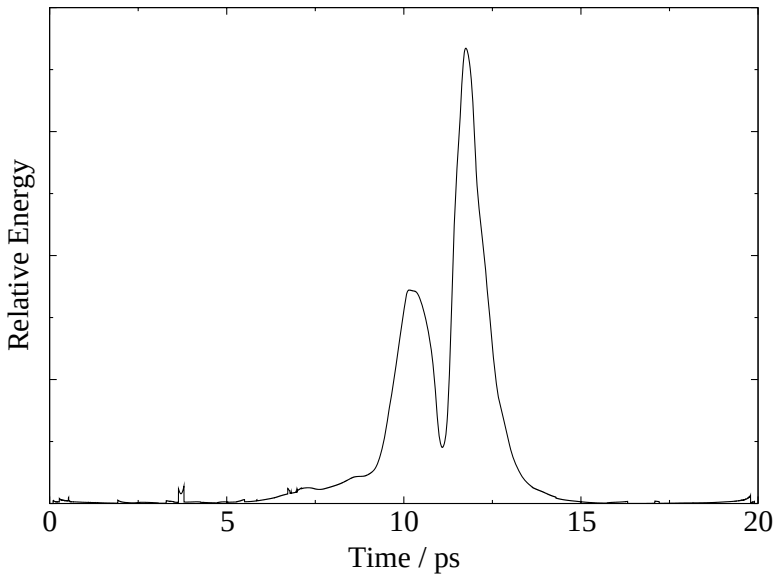

Figure 6.11. The relative amount of energy in the 0–55 cm^{-1} region of the pentane trajectory.

using a spectral density analysis of the ϕ and ψ dihedrals, but a 50 cm^{-1} amplifying filter failed to introduce much energy after a few applications (with a zero-step delay). When a 40 cm^{-1} filter was applied, energy was once again introduced into the system. Figure 6.12 shows the HHT plots of the dihedral angles from the original trajectory and the trajectory where the 50 cm^{-1} filter failed to work. The plots clearly show that the frequency of the important low-frequency mode has shifted down. Figure 6.13 shows the time-integrated frequency-energy plots for 0–70 cm^{-1} of the two trajectories. It is clear that not only has the majority of the energy shifted to lower frequency, but that

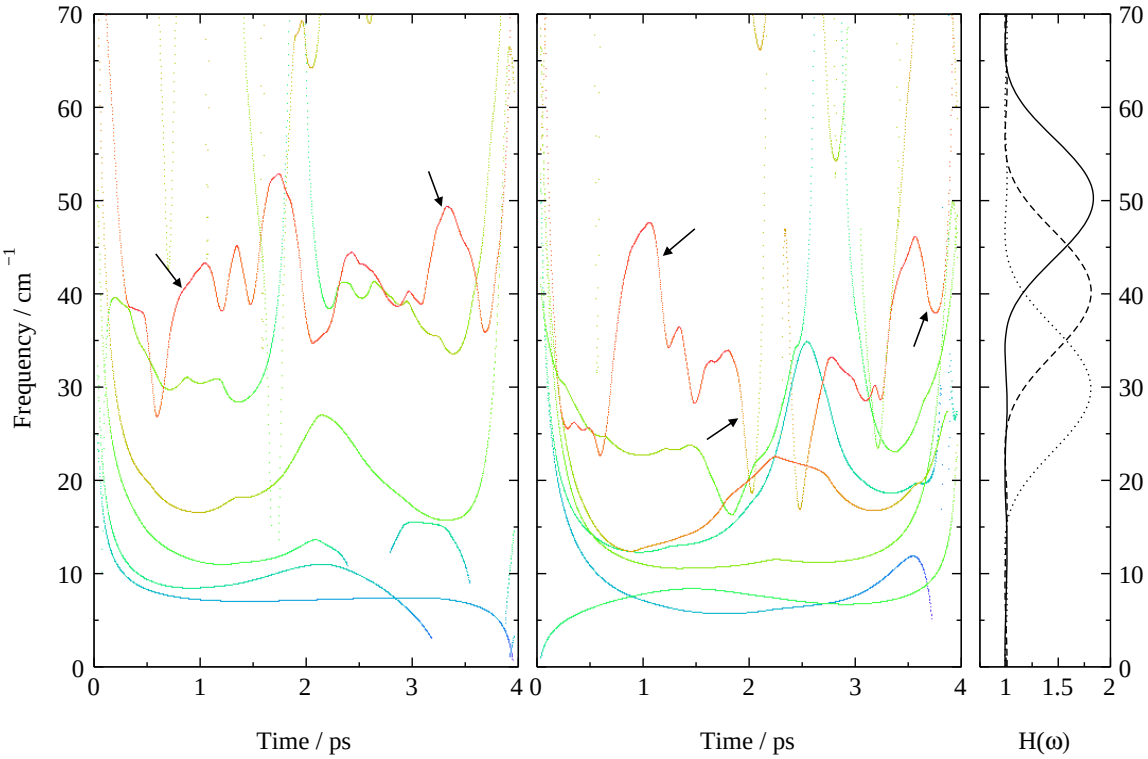

Figure 6.12. The HHT transforms of two alanine dipeptide trajectories using a logarithmic energy scale (left and middle) and the filters applied to the trajectories (right). Left: the initial trajectory, to which a $50\,\mathrm{cm}^{-1}$ amplifying filter was successfully applied. Right: the trajectory resulting from applying several $50\,\mathrm{cm}^{-1}$ amplifying filters. The important low-frequency IMF has been marked by arrows.

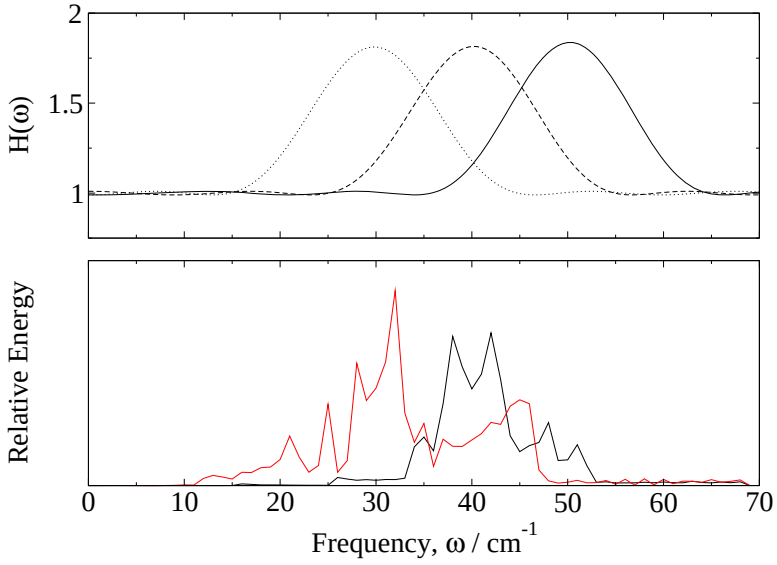

Figure 6.13. Top: the filters applied to the system. Bottom: the time integrals of the HHT transforms of figure 6.12 for 0–$70\,\mathrm{cm}^{-1}$. The black line is for the initial trajectory and the red line is for the enhanced trajectory.

the original filter ideally should have targeted $40\,\mathrm{cm}^{-1}$ and the following filter $30\,\mathrm{cm}^{-1}$.

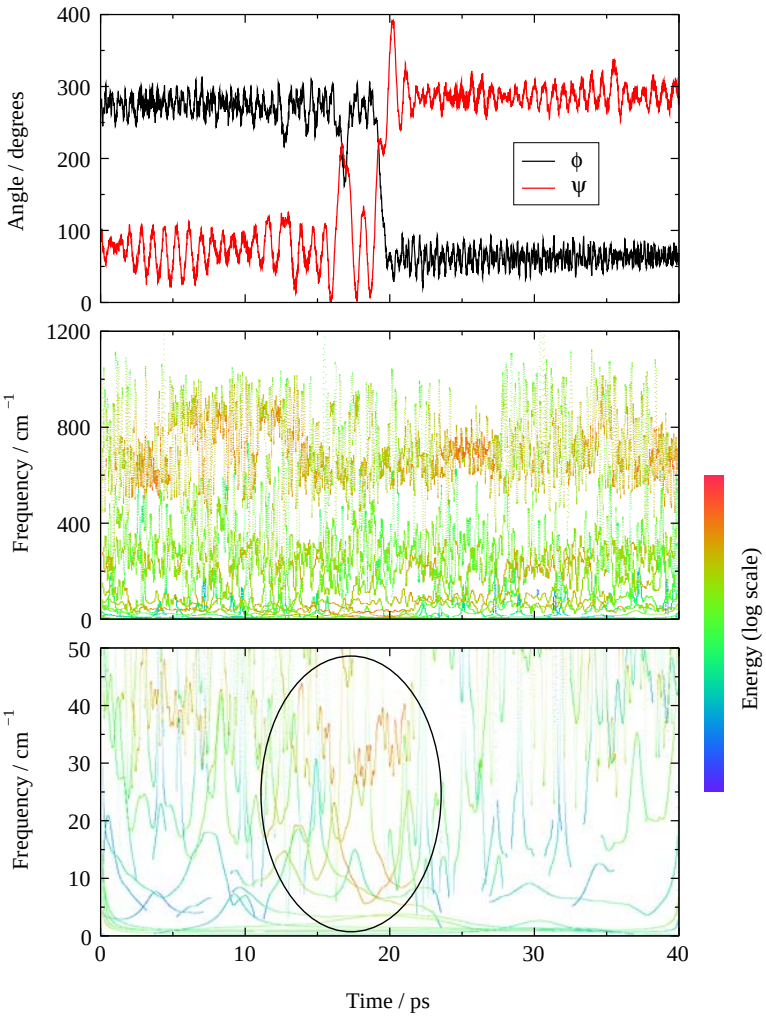

Figure 6.14. Top: the dihedral angle trajectory for a conformational transition in gas-phase alanine dipeptide. Middle: the HHT plot of the dihedral angles. Bottom: the 0–50 cm^{-1} region of the HHT plot. The increase in energy of the low-frequency IMFs during the transition period has been highlighted in the bottom figure.

The second analysis of alanine dipeptide is of the RDFMD simulation that caused a conformational transition by applying five 50 cm^{-1} and four 40 cm^{-1} amplifying filters. The trajectory has been extended both forwards and backwards in time and re-analysed using the HHT. The original trajectory is displayed in figure 5.16 (page 95) and the extended ϕ-ψ trajectory is shown in figure 6.14 along with the HHT plots of the dihedral angles for the whole frequency range and the 0–50 cm^{-1} range.

It is possible to see in the low-frequency HHT plot that additional low-frequency energy is present during the transition period (ringed in the figure). The time-energy

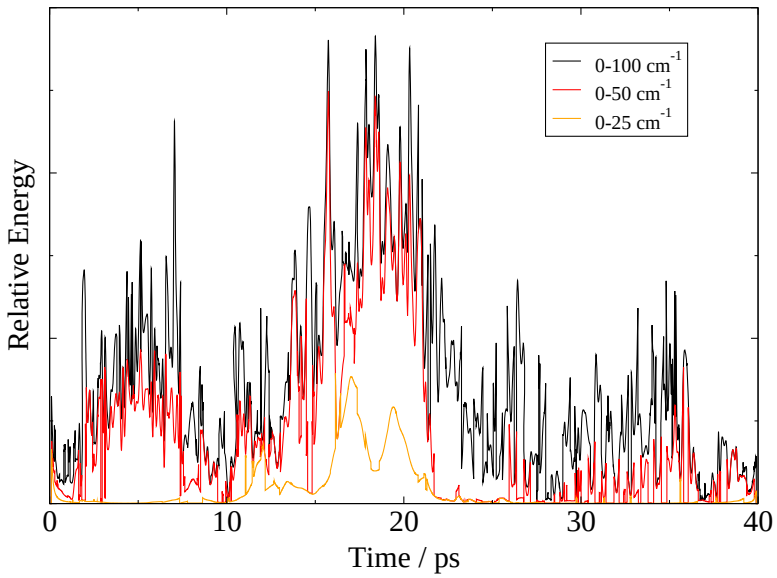

Figure 6.15. The relative amounts of energy in the 0–25 cm^{-1}, 0–50 cm^{-1} and 0–100 cm^{-1} frequency bands of alanine dipeptide during a conformational transition.

plots for three frequency bands have been calculated and are plotted in figure 6.15. In this more complicated case, the IMFs move into and out of each frequency band, causing discontinuities in the time-energy spectra. The 0–100 cm^{-1} and 0–50 cm^{-1} bands show a general increase in low-frequency energy either side of the filter application point of 20 ps, and the 0–25 cm^{-1} frequency band has two major peaks, one during the initial transition at approximately 19.5 ps and a second larger peak at approximately 17 ps when the molecule almost returns to the original conformation.

The HHT analysis method presents an opportunity to analyse the frequency and energy characteristics of a spontaneous transition and to compare the results with an RDFMD-induced transition. A spontaneous conformational transition occurred during an attempt to equilibrate the chloroform-solvated alanine dipeptide system at 293 K. The region around the transition point was analysed using the HHT method and the energy-time graph for 0–25 cm^{-1} is presented in figure 6.16 along with the same analysis of the RDFMD-induced transition presented earlier (figure 5.24). Both transition paths take the conformation from minimum B to minimum A along a path passing near to the β transition point. The induced transition occurs approximately five times faster than the natural transition (according to the peak widths in figure 6.16) but during both transitions energy moves into the low frequency modes.

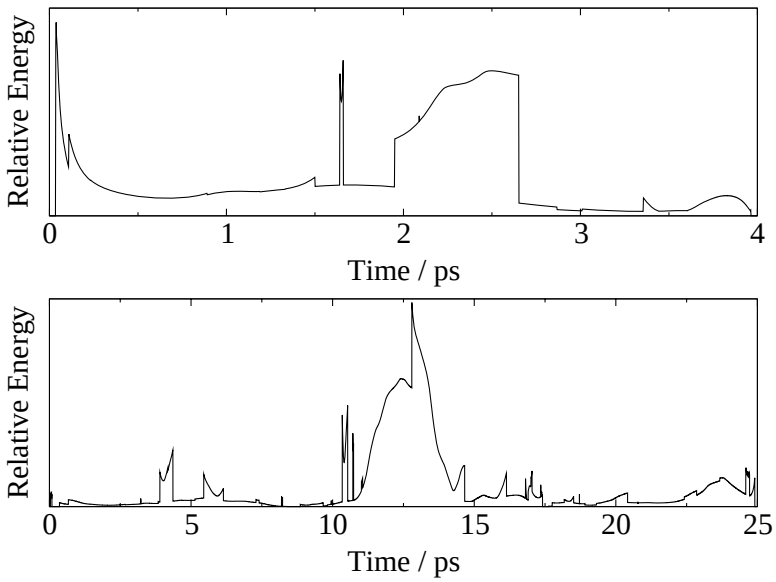

Figure 6.16. The time-energy plots of the 0–25 cm^{-1} frequency band for two conformational transitions in chloroform-solvated alanine dipeptide. Top: an RDFMD trajectory. Bottom: a spontaneously occurring event.

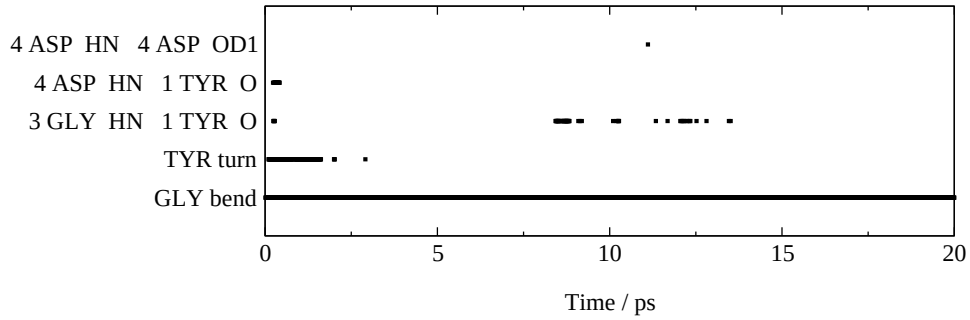

Figure 6.17. The secondary structure elements present during the 20 ns NPT YPGDV trajectory.

The final conformational transition to be re-analysed is part of the RDFMD simulation of YPGDV. The 19^{th} NPT simulation in the RDFMD series was recalculated, saving the coordinates every 10 fs. Figure 6.17 shows the variation in secondary structure and hydrogen bonding for this part of the simulation. The conformation starts with a turn at 1 TYR and hydrogen bonds between 3 GLY and 1 TYR, and 4 ASP and 1 TYR. The turn lasts until approximately 3 ps and does not re-appear during this portion of the simulation. Between approximately 6 ps and 14 ps the hydrogen bond between 3 GLY and 1 TYR re-appears.

The eight ϕ and ψ trajectories were processed through the EMD algorithm to produce 61 IMFs. The residual data after the IMFs were extracted are shown in figure

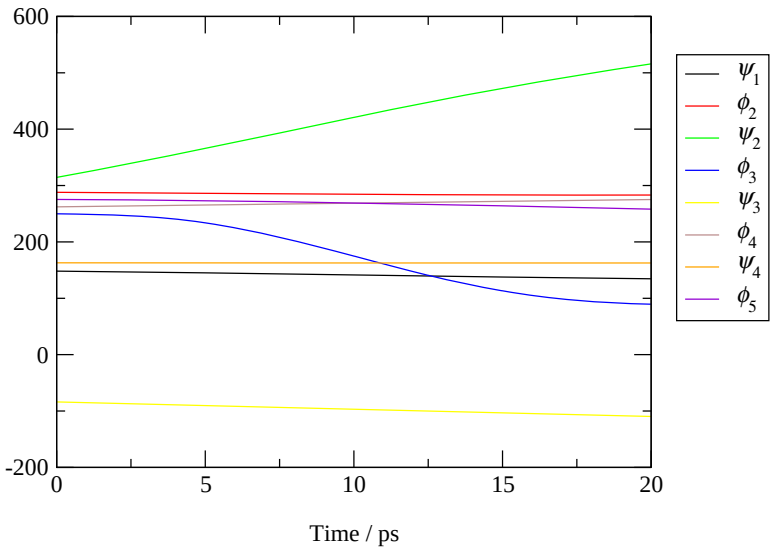

Figure 6.18. The residual data after the IMFs have been subtracted from the ϕ and ψ trajectories of a 20 ps NPT YPGDV trajectory.

6.18. The residual data effectively shows the trends in the dihedral angles. Figure 6.19 shows the ϕ-ψ trajectory and the Hilbert time-energy plots for the 0–25 cm^{-1} and 0–10 cm^{-1} frequency bands. The trend displayed in the residual data (figure 6.18) matches the trends evident in the original data (figure 6.19). Large peaks in the 0–25 cm^{-1} frequency band are present during the conformational change events (in angles ψ_2, ϕ_3, ψ_4 and ϕ_5), vindicating the choice of a 0–25 cm^{-1} amplifying filter for this system. Although this trajectory is part of the RDFMD trajectory, it is one of the NPT sections, and the conformational change events may be considered sufficiently distant in time from the filter applications to be spontaneous transitions.

6.5: Conclusion

The Fourier transform and spectral density have successfully been used in this thesis for identifying target frequencies. Care has been taken however, not too over-interpret the Fourier spectra and only data-sets where the system is in equilibrium (and the data-set is approximately stationary) have been analysed using the Fourier transform method. The spectrogram has some use but is limited by a trade-off between frequency and time resolution. The wavelet transform has very bad time resolution at the low frequencies of interest during conformational transitions. The HHT method has therefore been investigated.

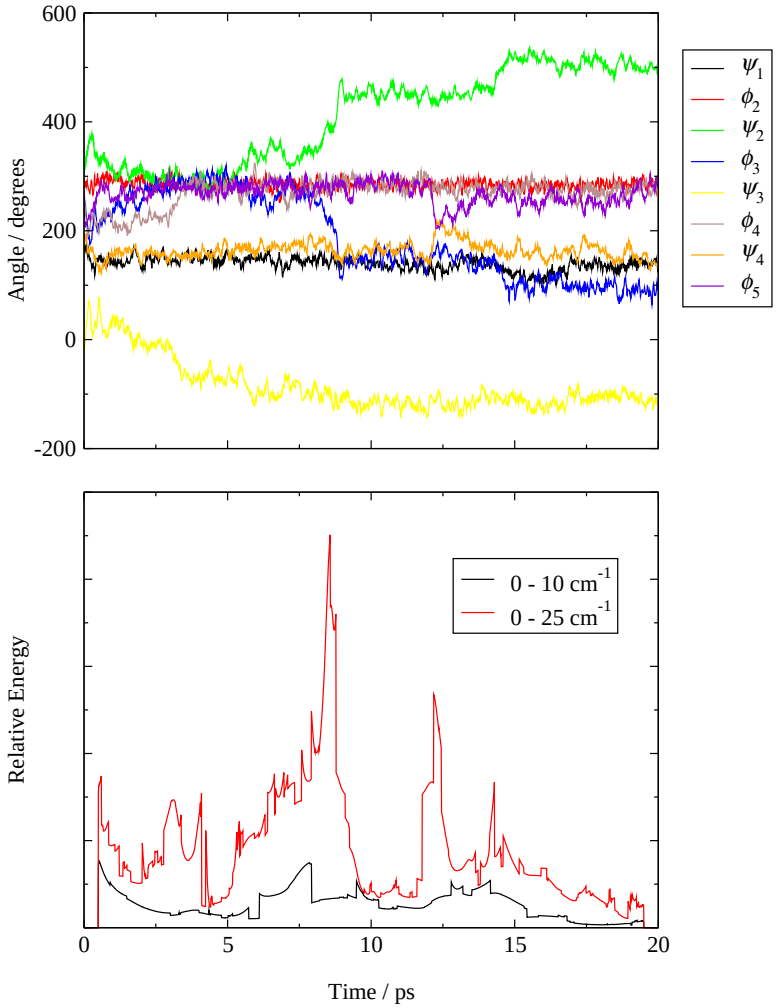

Figure 6.19. Top: the ϕ and ψ trajectory of an NPT section of the RDFMD YPGDV simulation. Bottom: the Hilbert time-energy spectra for two frequency bands.

Tests on the HHT method indicate that artifacts in the analysis are possible. Despite this, useful and consistent data has been obtained from the analysis of conformational transitions. The integration of the frequency axis to provide a time-energy spectrum during a transition is particularly useful. The analysis of five different conformational transitions provides a consistent picture of an increase in the energy of low-frequency oscillations during conformational transition events, vindicating the choices made in designing the amplifying filters for these systems. The observation that the frequency of a mode may shift in frequency as energy is introduced into the mode has also been confirmed.

Finally, the similarities between the low frequency energy profiles of induced and

spontaneous transitions validates the DFMD and RDFMD method of promoting conformational change by introducing energy into low frequency modes.

7: Conclusion

Molecular dynamics is a powerful method of simulating the dynamics of real systems on a computer. Unfortunately, the length of time it is possible to simulate is severely limited by the necessity of moving the system forward by very small time-steps. Currently, it is common to simulate a solvated protein system for a time of the order of nanoseconds.

The biochemical function of many protein systems is dependent on some form of conformational transition, and it is important to be able to simulate these transitions using molecular dynamics. The time-scales of these transitions, often of the order of milli-seconds, present a problem for normal MD simulations, which are unable to simulate for a sufficiently long time to adequately sample or even to see the transition events. Many methods have been proposed for modifying the basic MD method for the purpose of improving the study of this type of conformational transition. The methods may broadly be divided into three categories: generic methods to increase the amount of time simulated, methods that drive a simulation from one conformation to another conformation, and methods that drive conformational change without requiring knowledge of the final state. One method requiring no knowledge of the final state, the SEMD method, had been shown to drive conformational change in hexane by the application of a digital filter. In this thesis, the SEMD method has been greatly enhanced to become

the new method of Digitally Filtered Molecular Dynamics (DFMD).

DFMD involves the application of digital filters to modify the velocities in a molecular dynamics simulation. The modification is such that vibrations within the system are either enhanced or suppressed in a controlled fashion according to frequency. The application of this methodology to four test cases, water in the gas phase, butane in both gas and liquid phases, and the Syrian hamster prion protein, has been described, and in particular it has been shown that DFMD can bring about conformational change selectively. In the case of the prion protein, it has also been shown that DFMD is more efficient than the simpler alternative of non-selective heating.

While DFMD works very well in some systems, the method does have some limitations. It is difficult to target low-frequency oscillations in condensed phase systems where the energy rapidly dissipates from the target frequency. The requirement to introduce all the energy into the system in a single multiple-pulse filter application is also restrictive. The Reversible Digitally Filtered Molecular Dynamics (RDFMD) method improves on the DFMD method by applying the filter to trajectories sampled both forwards and backwards in time. This enables the user to choose exactly where in the trajectory the filter is applied. By choosing to space filter applications closely in the trajectory, one filter application may build on the energy input and enhanced trajectory of a previous application in a way that is not possible with DFMD. The RDFMD method thus provides a mechanism for introducing frequency-targeted energy into the system in a gradual manner.

The RDFMD method has been applied to single molecules of pentane and alanine dipeptide, solvated alanine dipeptide and solvated YPGDV. It was found that introducing energy into the system in a gradual manner could promote more reasonable transitions than introducing all the energy in one step. It was also shown that the introduction of extra energy into a vibrational mode could move the frequency of the vibration. The RDFMD method is able to adapt to these frequency shifts by the application of a series of different filters.

Successful conformational change events were induced in the gas phase systems and in the solvated alanine dipeptide model. Solvated alanine dipeptide was also used to show once again that placing energy into the system in a frequency-targeted manner

causes conformational change but placing the same amount of energy by unspecific heating does not.

The final system to be studied by RDFMD was the pentapeptide YPGDV solvated in water. A complex protocol for performing an NPT simulation interspersed with regular RDFMD filter applications was devised. The resulting trajectory was compared with a normal MD trajectory and with limited data from a simulation of YPGDV performed using the Self Guided Molecular Dynamics method (SGMD). The configurations generated during the RDFMD experiment had more defined secondary structure than those of the MD trajectory and fitted better with the experimental NOE data than either the MD or SGMD trajectories.

During the work on DFMD and RDFMD it became apparent that the spectral density analysis—the simulation equivalent of an IR spectrum—was too limited in its applicability and prone to artifacts. The Hilbert-Huang Transform (HHT), first published in 1998, provided the possibility of analysing the energy content of vibrational frequencies as a function of time. The HHT transform has been tested on some simple analytic data-sets and was found to have problems in some cases. However, its application to the analysis of transitions in the molecular dynamics simulations already presented proved to be invaluable. The HHT was able to confirm the shift in a mode's frequency as energy was introduced, and was able to show an increase in very low-frequency energy during transitions in five systems, including two transitions not induced by a digital filter but occurring spontaneously. The similarity between the low-frequency energy profiles of the induced and spontaneous transitions validates the use of low-frequency amplifying filters in the DFMD and RDFMD methods for enhancing conformational change.

The RDFMD method and the HHT analysis between them provide a powerful tool for inducing and analysing conformational transitions in complex molecules.

8: References

[1] W.L. Jorgensen, and J. Tirado-Rives, J. Am. Chem. Soc. **110**, 1657 (1988).

[2] W.L. Jorgensen, D.S. Maxwell, and J. Tirado-Rives, J. Am. Chem. Soc. **118**, 11225 (1996).

[3] W.D. Cornell, P. Cieplak, C.I. Bayly, I.R. Gould, K.M. Merz, D.M. Ferguson, D.C. Spellmeyer, T. Fox, J.W. Caldwell, and P.A. Kollman, J. Am. Chem. Soc. **118**, 2309 (1996).

[4] W.R.P. Scott, P.H. Hunenberger, I.G. Tironi, A.E. Mark, S.R. Billeter, J. Fennen, A.E. Torda, T. Huber, P. Kruger, and W.F. van Gunsteren, J. Phys. Chem. A **103**, 3596 (1999).

[5] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, and M. Karplus, J. Comput. Chem. **4**, 187 (1983).

[6] W.L. Jorgensen, BOSS 4.1, Yale University, New Haven, CT, 1999.

[7] W.L. Jorgensen, MCPRO 1.4, Yale University, New Haven, CT, 1996.

[8] J.-P. Ryckaert, G. Ciccoti, and H.J.C. Berendsen, J. Comput. Phys. **23**, 327 (1977).

[9] K.M. Pan, M. Baldwin, J. Nguyen, M. Gasset, A. Serban, D. Groth, I. Mehlhorn, Z.W. Huang, R.J. Fletterick, F.E. Cohen, and S.B. Prusiner, Proc. Natl. Acad. Sci.

USA **90**, 10962 (1993).

[10] J. Safar, P.P. Roller, D.C. Gajdusek, and C.J. Gibbs, Protein Sci. **2**, 2206 (1993).

[11] S.W. Rick, J.W. Erickson, and S.K. Burt, Proteins **32**, 7 (1998).

[12] C. Brandon, and J. Tooze, *Introduction to Protein Structure* (Garland Publishing Inc., New York, USA, 1991)

[13] L. Mirny, and E. Shakhnovich, Annu. Rev. Biophys. Biomol. Struct. **30**, 361 (2001).

[14] S. Doniach, and P. Eastman, Curr. Opin. Struct. Biol. **9**, 157 (1999).

[15] Y. Duan, and P.A. Kollman, Science **282**, 740 (1998).

[16] R.E. Bruccoleri, and M. Karplus, Biopolymers **29**, 1847 (1990).

[17] D.A. Case, D.A. Pearlman, J.W. Caldwell, T.E. Cheatham III, W.S. Ross, C.L. Simmerling, T.A. Darden, K.M. Merz, R.V. Stanton, A.L. Cheng, J.J. Vincent, M. Crowley, D.M. Ferguson, R.J. Radner, G.L. Seibel, U.C. Singh, P.K. Weiner, and P.A. Kollman, AMBER 5, University of California, San Francisco, 1997.

[18] T.R. Forester, and W. Smith, DL_POLY 2.9, CCLRC, Daresbury Laboratory, Warrington, England, 1997.

[19] S. Melchionna, and S. Cozzini, DLPROTEIN 1.2.

[20] M.E. Tuckerman, D.A. Yarne, S.O. Samuelson, A.L. Hughes, and G.J. Martyna, Comput. Phys. Comm. **128**, 333 (2000).

[21] D.M. York, T.A. Darden, and L.G. Pedersen, J. Chem. Phys. **99**, 8345 (1993).

[22] P. Ewald, Ann. Phys. **64**, 25 (1921).

[23] T. Darden, D. York, and L. Pedersen, J. Chem. Phys. **98**, 10089 (1993).

[24] U. Essmann, L. Perera, M.L. Berkowitz, T. Darden, H. Lee, and L.G. Pedersen, J. Chem. Phys. **103**, 8577 (1995).

[25] K.A. Feenstra, B. Hess, and H.J.C. Berendsen, J. Comput. Chem. **20**, 786 (1999).

[26] M.E. Tuckerman, G.J. Martyna, and B.J. Berne, J. Chem. Phys. **93**, 1287 (1990).

[27] M. Tuckerman, B.J. Berne, and G.J. Martyna, J. Chem. Phys. **97**, 1990 (1992).

[28] A.L. Cheng, and K.M. Merz, J. Phys. Chem. B **103**, 5396 (1999).

[29] T.C. Bishop, R.D. Skeel, and K. Schulten, J. Comput. Chem. **18**, 1785 (1997).

[30] H. Grubmuller, and P. Tavan, J. Comput. Chem. **19**, 1534 (1998).

[31] T. Schlick, E. Barth, and M. Mandziuk, Annu. Rev. Biophys. Biomol. Struct. **26**, (1997).

[32] R. Elber, and M. Karplus, Chem. Phys. Lett. **139**, 375 (1987).

[33] R. Czerminski, and R. Elber, J. Chem. Phys. **92**, 5580 (1990).

[34] M.A. Ech-Cherif El-Kettani, and J. Durup, Biopolymers **32**, 561 (1992).

[35] O.S. Smart, Chem. Phys. Lett. **222**, 503 (1994).

[36] O.S. Smart, and J.M. Goodfellow, Mol. Simul. **14**, 291 (1995).

[37] S. Fischer, and M. Karplus, Chem. Phys. Lett. **194**, 252 (1992).

[38] J. Schlitter, M. Engels, and P. Kruger, J. Mol. Graphics **12**, 84 (1994).

[39] J. Schlitter, M. Engels, P. Kruger, E. Jacoby, and A. Wollmer, Mol. Simul. **10**, 291 (1993).

[40] E. Jacoby, P. Kruger, J. Schlitter, D. Roper, and A. Wollmer, Protein Eng. **9**, 113 (1996).

[41] J.F. Diaz, B. Wroblowski, J. Schlitter, and Y. Engelborghs, Proteins **28**, 434 (1997).

[42] B. Wroblowski, J.F. Diaz, J. Schlitter, and Y. Engelborghs, Protein Eng. **10**, 1163 (1997).

[43] P. Ferrara, J. Apostolakis, and A. Caflisch, J. Phys. Chem. B **104**, 4511 (2000).

[44] P. Ferrara, J. Apostolakis, and A. Caflisch, Proteins **39**, 252 (2000).

[45] C. Guilbert, D. Perahia, and L. Mouawad, Comput. Phys. Comm. **91**, 263 (1995).

[46] J.S.D. Santos, S. Fischer, C. Guilbert, A. Lewit-Bentley, and J.C. Smith, Biochemistry **39**, 14065 (2000).

[47] M. Marchi, and P. Ballone, J. Chem. Phys. **110**, 3697 (1999).

[48] S.C. Harvey, and H.A. Gabb, Biopolymers **33**, 1167 (1993).

[49] R. Olender, and R. Elber, J. Chem. Phys. **105**, 9299 (1996).

[50] R. Elber, J. Meller, and R. Olender, J. Phys. Chem. B **103**, 899 (1999).

[51] B. Isralewitz, S. Izrailev, and K. Schulten, Biophys. J. **73**, 2972 (1997).

[52] S.J. Marrink, O. Berger, P. Tieleman, and F. Jahnig, Biophys. J. **74**, 931 (1998).

[53] H. Lu, B. Isralewitz, A. Krammer, V. Vogel, and K. Schulten, Biophys. J. **75**, 662 (1998).

[54] M. Levitt, C. Sander, and P.S. Stern, J. Mol. Biol. **181**, 423 (1985).

[55] A. Amadei, A.B.M. Linssen, and H.J.C. Berendsen, Proteins **17**, 412 (1993).

[56] P. Dauber-Osguthorpe, C.M. Maunder, and D.J. Osguthorpe, J. Comput.-Aided Mol. Design **10**, 177 (1996).

[57] X.W. Wu, and S.M. Wang, J. Phys. Chem. B **102**, 7238 (1998).

[58] X.W. Wu, and S.M. Wang, J. Chem. Phys. **110**, 9401 (1999).

[59] X.W. Wu, and S.M. Wang, J. Phys. Chem. B **105**, 2227 (2001).

[60] X.W. Wu, and S.M. Wang, J. Phys. Chem. B **104**, 8023 (2000).

[61] W. Shinoda, and M. Mikami, Chem. Phys. Lett. **335**, 265 (2001).

[62] H. Grubmuller, Phys. Rev. E **52**, 2893 (1995).

[63] B.G. Schulze, H. Grubmuller, and J.D. Evanseck, J. Am. Chem. Soc. **122**, 8700 (2000).

[64] R. Abseher, and M. Nilges, Proteins **39**, 82 (2000).

[65] A. Amadei, A.B.M. Linssen, B.L. de Groot, D.M.F. van Aalten, and H.J.C. Berendsen, J. Biomol. Struct. Dyn. **13**, 615 (1996).

[66] B.L. de Groot, A. Amadei, D.M.F. van Aalten, and H.J.C. Berendsen, J. Biomol. Struct. Dyn. **13**, 741 (1996).

[67] J. Kleinjung, P. Bayley, and F. Fraternali, Febs Lett. **470**, 257 (2000).

[68] B.L. de Groot, D.M.F. van Aalten, R.M. Scheek, A. Amadei, G. Vriend, and H.J.C. Berendsen, Proteins **29**, 240 (1997).

[69] R. Elber, J. Chem. Phys. **93**, 4312 (1990).

[70] W.M. Zheng, and Q. Zheng, J. Chem. Phys. **106**, 1191 (1997).

[71] C. Simmerling, J.L. Miller, and P.A. Kollman, J. Am. Chem. Soc. **120**, 7149 (1998).

[72] M.A. Balsera, W. Wriggers, Y. Oono, and K. Schulten, J. Phys. Chem. **100**, 2567 (1996).

[73] R.B. Sessions, P. Dauber-Osguthorpe, and D.J. Osguthorpe, J. Mol. Biol. **210**, 617 (1989).

[74] P. Dauber-Osguthorpe, and D.J. Osguthorpe, J. Am. Chem. Soc. **112**, 7921 (1990).

[75] R.B. Sessions, D.J. Osguthorpe, and P. Dauber-Osguthorpe, J. Phys. Chem. **99**, 9034 (1995).

[76] P. Dauber-Osguthorpe, and D.J. Osguthorpe, Biochemistry **29**, 8223 (1990).

[77] D.J. Osguthorpe, and P. Dauber-Osguthorpe, J. Mol. Graphics **10**, 178 (1992).

[78] P. Dauber-Osguthorpe, and D.J. Osguthorpe, J. Comput. Chem. **14**, 1259 (1993).

[79] A.P. Lemon, P. Dauber-Osguthorpe, and D.J. Osguthorpe, Comput. Phys. Comm. **91**, 97 (1995).

[80] M. Levitt, J. Mol. Biol. **220**, 1 (1991).

[81] C.S. Williams, *Designing Digital Filters* (Prentice-Hall, Englewood Cliffs, N.J., USA, 1986)

[82] M. Frigo, and S.G. Johnson, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing **3**, 1381 (1998).

[83] O. Teleman, B. Jonsson, and S. Engstrom, Mol. Phys. **60**, 193 (1987).

[84] S.J. Weiner, P.A. Kollman, D.A. Case, U.C. Singh, C. Ghio, G. Alagona, S. Profeta, and P. Weiner, J. Am. Chem. Soc. **106**, 765 (1984).

[85] W.L. Jorgensen, J.D. Madura, and C.J. Swenson, J. Am. Chem. Soc. **106**, 6638 (1984).

[86] L.G. Goldfarb, R.B. Petersen, M. Tabaton, P. Brown, A.C. Leblanc, P. Montagna, P. Cortelli, J. Julien, C. Vital, W.W. Pendelbury, M. Haltia, P.R. Wills, J.J. Hauw, P.E. McKeever, L. Monari, B. Schrank, G.D. Swergold, L. Autiliogambetti, D.C. Gajdusek, E. Lugaresi, and P. Gambetti, Science **258**, 806 (1992).

[87] O.G. Parchment, and J.W. Essex, Proteins **38**, 327 (2000).

[88] W.L. Jorgensen, J. Chandrasekhar, J.D. Madura, R.W. Impey, and M.L. Klein, J. Chem. Phys. **79**, 926 (1983).

[89] W.D. Cornell, P. Cieplak, C.I. Bayly, I.R. Gould, K.M. Merz, D.M. Ferguson, D.C. Spellmeyer, T. Fox, J.W. Caldwell, and P.A. Kollman, J. Am. Chem. Soc. **117**, 5179 (1995).

[90] S.C. Phillips, J.W. Essex, and C.M. Edge, J. Chem. Phys. **112**, 2586 (2000).

[91] M.P. Allen, and D.J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, Oxford, UK, 1987)

[92] G.J. Martyna, M.E. Tuckerman, D.J. Tobias, and M.L. Klein, Mol. Phys. **87**, 1117 (1996).

[93] The MathWorks, Inc., MATLAB 5.3.0, 1984–1999

[94] B.M. Pettitt, and M. Karplus, Chem. Phys. Lett. **121**, 194 (1985).

[95] T.J. Marrone, M.K. Gilson, and J.A. McCammon, J. Phys. Chem. **100**, 1439 (1996).

[96] C. Bartels, and M. Karplus, J. Comput. Chem. **18**, 1450 (1997).

[97] M. Scarsi, J. Apostolakis, and A. Caflisch, J. Phys. Chem. B **102**, 3637 (1998).

[98] P.E. Smith, J. Chem. Phys. **111**, 5568 (1999).

[99] A.G. Anderson, and J. Hermans, Proteins **3**, 262 (1988).

[100] T. Lazaridis, D.J. Tobias, C.L. Brooks, and M.E. Paulaitis, J. Chem. Phys. **95**, 7612 (1991).

[101] J. Apostolakis, P. Ferrara, and A. Caflisch, J. Chem. Phys. **110**, 2099 (1999).

[102] W.G. Hoover, Phys. Rev. A **31**, 1695 (1985).

[103] V. Madison, and K.D. Kopple, J. Am. Chem. Soc. **102**, 4855 (1980).

[104] W.L. Jorgensen, J.M. Briggs, and M.L. Contreras, J. Phys. Chem. **94**, 1683 (1990).

[105] J.W Essex, D.Phil thesis, Oriel College, Oxford, 1992.

[106] S. Melchionna, G. Ciccotti, and B.L. Holian, Mol. Phys. **78**, 533 (1993).

[107] H.J. Dyson, M. Rance, R.A. Houghten, R.A. Lerner, and P.E. Wright, J. Mol. Biol. **201**, 161 (1988).

[108] D.J. Tobias, J.E. Mertz, and C.L. Brooks, Biochemistry **30**, 6054 (1991).

[109] M.E. Karpen, D.J. Tobias, and C.L. Brooks, Biochemistry **32**, 412 (1993).

[110] A.D. Mackerell, D. Bashford, M. Bellott, R.L. Dunbrack, J.D. Evanseck, M.J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-mccarthy, L. Kuchnir, K. Kuczera, F.T.K. Lau, C. Mattos, S. Michnick, T. Ngo, D.T. Nguyen, B. Prodhom, W.E. Reiher, B. Roux, M. Schlenkrich, J.C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus, J. Phys. Chem. B **102**, 3586 (1998).

[111] R. Koradi, M. Billeter, and K. Wüthrich, J. Mol. Graphics **14**, 51 (1996).

[112] W. Kabsch, and C. Sander, Biopolymers **22**, 2577 (1983).

[113] A.V. Oppenheim, and R.W. Schafer, *Digital Signal Processing* (Prentice Hall, Englewood Cliffs, N.J., USA, 1989)

[114] C. Torrence, and G.P. Compo, Bull. Amer. Meteorol. Soc. **79**, 61 (1998).

[115] B.E. Usevitch, IEEE Signal Process. Mag. **18**, 22 (2001).

[116] J.S. Bendat, *The Hilbert Transform* (Brüel & Kjær, Nærum, Denmark).

[117] W. Huang, Z. Shen, N.E. Huang, and Y.C. Fung, Proc. Natl. Acad. Sci. U.S.A. **95**, 4816 (1998).

[118] J.C. Echeverria, J.A. Crowe, M.S. Woolfson, and B.R. Hayes-Gill, Med. Biol. Eng. Comput. **39**, 471 (2001).

[119] K.Y. Chen, H.C. Yeh, S.Y. Su, C.H. Liu, and N.E. Huang, Geophys. Res. Lett. **28**, 3107 (2001).

[120] R.W. Komm, F. Hill, and R. Howe, Astrophys. J. **558**, 428 (2001).

Appendix A: Proofs

A.1: Spectral Density in Terms of Coordinate

If we consider a single particle (e.g. an atom) of mass m, at position x, moving as a simple harmonic oscillator with force constant k, then

$$F(x) = -kx \tag{A.1}$$

Putting this into $F = ma$ gives:

$$-kx = m\ddot{x} \tag{A.2}$$

We define the angular frequency, ω, as

$$\omega^2 = \frac{k}{m} \tag{A.3}$$

Then putting equation A.3 into equation A.2 and using standard differential equation results we obtain:

$$\begin{aligned} &\ddot{x} + \omega^2 x = 0 \\ &\Rightarrow x(t) = A\sin(\omega t - \delta) \\ &\Rightarrow \dot{x}(t) = A\omega\cos(\omega t - \delta) \end{aligned} \tag{A.4}$$

where A is the amplitude of the motion and δ is the phase.

The kinetic energy of the particle, T, is

$$\begin{aligned} T &= \frac{1}{2}mv^2 \\ &= \frac{1}{2}m\dot{x}^2(t) \\ &= \frac{1}{2}mA^2\omega^2\cos^2(\omega t-\delta) \\ &= \frac{1}{2}kA^2\cos^2(\omega t-\delta) \\ &= \frac{1}{4}kA^2\left(1+\cos(2\omega t-2\delta)\right) \end{aligned} \tag{A.5}$$

For the potential energy, U, we calculate the work, W, done to displace the particle by distance x:

$$dW = -F\,dx = kx\,dx \tag{A.6}$$

Integrating from 0 to x and setting the potential energy equal to the work done on the particle, we have

$$\begin{aligned} U &= \int dW = \int_0^x kx\,dx = \left[\frac{1}{2}kx^2\right]_0^x = \frac{1}{2}kx^2 \\ &= \frac{1}{2}kA^2\sin^2(\omega t-\delta) \\ &= \frac{1}{4}kA^2\left(1-\cos(2\omega t-2\delta)\right) \end{aligned} \tag{A.7}$$

The total energy, E, is then

$$\begin{aligned} E = T+U &= \frac{1}{2}kA^2[\cos^2(\omega t-\delta)+\sin^2(\omega t-\delta)] \\ &= \frac{1}{2}kA^2 \end{aligned} \tag{A.8}$$

Equation A.7 can be rewritten in terms of the maximum displacement $x_{\max}$:

$$E = T+U = \frac{1}{2}kx_{\max}^2 \tag{A.9}$$

The simple harmonic oscillator has only one frequency of vibration but we can consider a more general case of equation A.9 where the total energy is made up of contributions from a continuous distribution of frequencies each with its own force constant,

$k(\nu)$. This is facilitated by the introduction of a Fourier transform:

$$\begin{aligned} E(\nu) &= \frac{1}{2}k(\nu)x^2_{\max}(\nu) \\ &= \frac{1}{2}k(\nu)\left(\int_{-\infty}^{\infty} x(t)\mathrm{e}^{-2\pi i\nu t}\,dt\right)^2 \end{aligned} \tag{A.10}$$

By putting $\omega = 2\pi\nu$ into equation A.3 (where ω is in $\mathrm{rad\,s^{-1}}$ and ν is in Hz) we can get an expression for k in terms of ν:

$$\nu = \frac{1}{2\pi}\sqrt{\frac{k}{m}} \qquad \text{or} \qquad k = (2\pi\nu)^2 m \tag{A.11}$$

Substituting equation A.11 into A.10:

$$E(\nu) = \frac{1}{2}(2\pi\nu)^2 m\left(\int_{-\infty}^{\infty} x(t)\mathrm{e}^{-2\pi i\nu t}\,dt\right)^2 \tag{A.12}$$

Dividing equation A.12 through by $\frac{1}{2}k_{\mathrm{B}}T$ (where k_{B} is the Bolztman constant and T is the temperature) makes it dimension-less. We call this quantity $g(\nu)$:

$$g(\nu) = \frac{1}{k_{\mathrm{B}}T}m\left(2\pi\nu\int_{-\infty}^{\infty} x(t)\mathrm{e}^{-2\pi i\nu t}\,dt\right)^2 \tag{A.13}$$

The integral of $g(\nu)$ is the number of degrees of freedom in the system.

For a system of n particles, we can use a more general equation:

$$g(\nu) = \sum_n g_n(\nu) = \sum_n \frac{1}{k_{\mathrm{B}}T}m_n\left(2\pi\nu\int_{-\infty}^{\infty} x_n(t)\mathrm{e}^{-2\pi i\nu t}\,dt\right)^2 \tag{A.14}$$

A.2: Spectral Density in Terms of Velocity

To derive this we use integration by parts on the integral from equation A.14:

$$\begin{aligned} \int_{-\infty}^{\infty} x_n(t)\mathrm{e}^{-2\pi i\nu t}\,dt &= \left[x_n(t)\left(\frac{1}{-2\pi i\nu}\right)\mathrm{e}^{-2\pi i\nu t}\right]_{-\infty}^{\infty} - \int_{-\infty}^{\infty}\dot{x}_n(t)\left(\frac{1}{-2\pi i\nu}\right)\mathrm{e}^{-2\pi i\nu t}\,dt \\ &= \left(\frac{1}{2\pi i\nu}\right)\int_{-\infty}^{\infty}\dot{x}_n(t)\mathrm{e}^{-2\pi i\nu t}\,dt \end{aligned} \tag{A.15}$$

Inserting equation A.15 in equation A.14 gives:

$$\begin{aligned} g_n(\nu) &= \frac{1}{k_{\mathrm{B}}T} m_n \left(\frac{1}{i} \int_{-\infty}^{\infty} \dot{x}_n(t) \mathrm{e}^{-2\pi i \nu t} \, dt \right)^2 \\ &= \frac{1}{k_{\mathrm{B}}T} m_n \left(\int_{-\infty}^{\infty} \dot{x}_n(t) \mathrm{e}^{-2\pi i \nu t} \, dt \right)^2 \end{aligned} \tag{A.16}$$

The i can be removed from the equation because the FT is complex and is being multiplied by its complex conjugate.

A.3: Fourier Filter Design Equation

The following proof obtains the design equation for the Fourier filter design method. $\mathcal{D}(\omega)$ is the desired filter response and $\mathcal{H}(\omega)$ is the actual response. The error function that the Fourier design method minimises, $\mathcal{E}$, is

$$\mathcal{E} = \int_{-\pi}^{\pi} |\mathcal{D}(\omega) - \mathcal{H}(\omega)|^2 \, d\omega \tag{A.17}$$

The error function must be minimised with respect to all the filter coefficients, c_n:

$$\begin{aligned} \text{Let} \quad & \frac{d\mathcal{E}}{dc_n} = 0, \qquad \forall n \\ \text{Then} \quad 0 &= \frac{d}{dc_n} \int_{-\pi}^{\pi} |\mathcal{D}(\omega) - \mathcal{H}(\omega)|^2 \, d\omega \\ &= \frac{d}{dc_n} \int_{-\pi}^{\pi} [\mathcal{D}(\omega) - \mathcal{H}(\omega)] \, [\mathcal{D}(\omega) - \mathcal{H}(\omega)]^* \, d\omega \\ &= \int_{-\pi}^{\pi} \frac{d}{dc_n} \left([\mathcal{D}(\omega) - \mathcal{H}(\omega)] \, [\mathcal{D}(\omega) - \mathcal{H}(\omega)]^* \right) d\omega \end{aligned} \tag{A.18}$$

Then using the chain rule of differentiation:

$$\begin{aligned} \frac{d\mathcal{E}}{dc_n} = & \int_{-\pi}^{\pi} \left(\frac{d}{dc_n} [\mathcal{D}(\omega) - \mathcal{H}(\omega)] \right) . \, [\mathcal{D}(\omega) - \mathcal{H}(\omega)]^* \, d\omega \\ & + \int_{-\pi}^{\pi} [\mathcal{D}(\omega) - \mathcal{H}(\omega)] \, . \left(\frac{d}{dc_n} [\mathcal{D}(\omega) - \mathcal{H}(\omega)]^* \right) d\omega \end{aligned} \tag{A.19}$$

The two parts of the sum are complex conjugates so when one is $a + ib$ the other is $a - ib$. The sum is $2a$ and will only be zero if the real part is zero. A solution is to have

the whole of each term zero. When one term is zero the other must also be zero so we will consider just the second term:

$$\int_{-\pi}^{\pi} [\mathcal{D}(\omega) - \mathcal{H}(\omega)] \cdot \left(\frac{d}{dc_n} [\mathcal{D}(\omega) - \mathcal{H}(\omega)]^* \right) d\omega \tag{A.20}$$

The second term of equation A.20 can be expanded:

$$\frac{d}{dc_n} [\mathcal{D}(\omega) - \mathcal{H}(\omega)]^* = \frac{d}{dc_n} [\mathcal{D}^*(\omega)] - \frac{d}{dc_n} [\mathcal{H}^*(\omega)] \tag{A.21}$$

The differential of $\mathcal{D}^*(\omega)$ with respect to c_n is zero as the desired response is fixed and not dependent on the coefficients. The differential of $\mathcal{H}^*(\omega)$ can be found introducing $\mathcal{H}^*(\omega)$ as a function of c_n (see equation 3.15):

$$\frac{d}{dc_n} [\mathcal{H}^*(\omega)] = \frac{d}{dc_n} \left(\sum_{k=-\infty}^{\infty} c_k \mathrm{e}^{i\omega k} \right) = \mathrm{e}^{i\omega n} \tag{A.22}$$

Putting this back into equation A.20 gives:

$$\begin{aligned} \int_{-\pi}^{\pi} [\mathcal{D}(\omega) - \mathcal{H}(\omega)] \mathrm{e}^{i\omega n} \, d\omega &= 0 \\ \int_{-\pi}^{\pi} \mathcal{D}(\omega) \mathrm{e}^{i\omega n} - \int_{-\pi}^{\pi} \mathcal{H}(\omega) \mathrm{e}^{i\omega n} \, d\omega &= 0 \\ \int_{-\pi}^{\pi} \mathcal{D}(\omega) \mathrm{e}^{i\omega n} &= \int_{-\pi}^{\pi} \mathcal{H}(\omega) \mathrm{e}^{i\omega n} \, d\omega \end{aligned} \tag{A.23}$$

Replacing $\mathcal{H}(\omega)$ with its expanded form we obtain:

$$\begin{aligned} \int_{-\pi}^{\pi} \mathcal{D}(\omega) \mathrm{e}^{i\omega n} &= \int_{-\pi}^{\pi} \mathcal{H}(\omega) \mathrm{e}^{i\omega n} \, d\omega \\ &= \int_{-\pi}^{\pi} \left(\sum_{k=-\infty}^{\infty} c_k \mathrm{e}^{-i\omega k} \right) \mathrm{e}^{i\omega n} \, d\omega \\ &= \int_{-\pi}^{\pi} \sum_{k=-\infty}^{\infty} c_k \mathrm{e}^{-i\omega(k-n)} \, d\omega \\ &= \sum_{k=-\infty}^{\infty} c_k \int_{-\pi}^{\pi} \mathrm{e}^{-i\omega(k-n)} \, d\omega \end{aligned} \tag{A.24}$$

Using the Euler formula $\exp(-i\theta) = \cos\theta - i\sin\theta$ the integral becomes:

$$\int_{-\pi}^{\pi} \mathrm{e}^{-i\omega(k+n)} d\omega = \int_{-\pi}^{\pi} \cos(\omega(k-n)) \, d\omega - i \int_{-\pi}^{\pi} \sin(\omega(k-n)) \, d\omega \tag{A.25}$$

If $k - n \neq 0$ then both of the trigonometric terms are zero as we are integrating over an integral number of periods. If $k - n = 0$ then the cosine integral is 2π and the sine integral is zero:

$$\int_{-\pi}^{\pi} \mathrm{e}^{-i\omega(k+n)} d\omega = \begin{cases} 2\pi, & \text{if } k = n; \\ 0, & \text{otherwise.} \end{cases} \tag{A.26}$$

Putting this result back into equation A.24 we have an equation just in terms of $\mathcal{D}(\omega)$ and c_n:

$$\begin{aligned} \int_{-\pi}^{\pi} \mathcal{D}(\omega) \mathrm{e}^{i\omega n} \, d\omega &= 2\pi c_n \\ \text{or} \quad c_n &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{D}(\omega) \mathrm{e}^{i\omega n} \, d\omega \end{aligned} \tag{A.27}$$

A.4: Periodic Convolution

If we have coefficient sets a_n and b_n and we let $c_n = a_n b_n$ then the frequency response of c_n is the periodic convolution of the responses of a_n and b_n.

$$\begin{aligned} \mathcal{H}_c(\omega) &= \sum_{n=-\infty}^{\infty} c_n \mathrm{e}^{-i\omega n} \\ &= \sum_{n=-\infty}^{\infty} a_n b_n \mathrm{e}^{-i\omega n} \end{aligned} \tag{A.28}$$

In order to proceed with this derivation we rewrite b_n in terms of a sum and an integral of exponents:

$$\text{Let} \quad b_n = \frac{1}{2\pi} \sum_{m=-\infty}^{\infty} \left(b_n \int_{-\pi}^{\pi} \mathrm{e}^{i(n-m)s} \, ds \right) \tag{A.29}$$

The integral is an integral of sines and cosines over an integer number of periods and evaluates to zero or 2π:

$$\int_{-\pi}^{\pi} \mathrm{e}^{i(n-m)s} \, ds = \begin{cases} 2\pi, & \text{if } n = m; \\ 0, & \text{otherwise.} \end{cases} \tag{A.30}$$

Putting this representation of b_n back into equation A.28 gives:

$$\begin{aligned}
\mathcal{H}_c(\omega) &= \sum_{n=-\infty}^{\infty}\left(a_n\left[\frac{1}{2\pi}\sum_{m=-\infty}^{\infty}\left(b_n\int_{-\pi}^{\pi}\mathrm{e}^{i(n-m)s}\,ds\right)\right]\mathrm{e}^{-i\omega n}\right) \\
&= \frac{1}{2\pi}\sum_{n=-\infty}^{\infty}\left(a_n\mathrm{e}^{-i\omega n}\sum_{m=-\infty}^{\infty}\left(b_m\int_{-\pi}^{\pi}\mathrm{e}^{i(n-m)s}\,ds\right)\right) \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi}\sum_{n=-\infty}^{\infty}\left(a_n\mathrm{e}^{-i\omega n}\sum_{m=-\infty}^{\infty}b_m\mathrm{e}^{i(n-m)s}\right)ds \qquad (A.31) \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\sum_{n=-\infty}^{\infty}a_n\mathrm{e}^{-i\omega n}\mathrm{e}^{ins}\right]\left[\sum_{m=-\infty}^{\infty}b_m\mathrm{e}^{-ims}\right]ds \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\sum_{n=-\infty}^{\infty}a_n\mathrm{e}^{-i(\omega-s)n}\right]\left[\sum_{m=-\infty}^{\infty}b_m\mathrm{e}^{-ims}\right]ds
\end{aligned}$$

The final terms in equation A.31 are frequency response terms:

$$\mathcal{H}_c(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\mathcal{H}_a(\omega-s)\mathcal{H}_b(s)\,ds \qquad (A.32)$$

A.5: Phase Change on the First Filter Pulse Application

Let us assume we have simple harmonic oscillator, equilibrium position r_0, amplitude α, frequency $\omega\,\mathrm{rad\,s^{-1}}$, phase δ and position r at time t:

$$r = r_0 + \alpha\sin(\omega t - \delta) \qquad (A.33)$$

The potential energy, U, is then:

$$\begin{aligned}
U(t) &= \frac{1}{2}k(r-r_0)^2 \\
&= \frac{1}{2}k\alpha^2\sin^2(\omega t-\delta) \qquad (A.34) \\
&= \frac{1}{4}k\alpha^2[1-cos(2\omega t-2\delta)]
\end{aligned}$$

where k is the force constant of the oscillation. The total energy, E, is constant and is the sum of U and the kinetic energy T. The total energy must be equal to the maximum

value of U which is $k\alpha^2/2$. We can then derive the kinetic energy:

$$\begin{aligned} T(t) &= E - U(t) \\ &= \frac{1}{2}k\alpha^2 - \frac{1}{4}k\alpha^2[1 - cos(2\omega t - 2\delta)] \\ &= \frac{1}{4}k\alpha^2[1 + cos(2\omega t - 2\delta)] \end{aligned} \tag{A.35}$$

An equation for δ can then be formed from equation A.35:

$$\begin{aligned} T(t) &= \frac{1}{4}k\alpha^2[1 + cos(2\omega t - 2\delta)] \\ \frac{4T(t)}{k\alpha^2} - 1 &= cos(2\omega t - 2\delta) \\ \frac{4T(t)}{2E} - 1 &= cos(2\omega t - 2\delta) \\ \delta &= \frac{1}{2}\left[2\omega t - \cos^{-1}\left(\frac{T-U}{E}\right)\right] \end{aligned} \tag{A.36}$$

On application, at time $t = t_0$, of the first pulse of a filter with $A = [\mathcal{H}(\omega)]^2$, the kinetic energy goes to T_1 and the potential energy to U_1:

$$\begin{aligned} T_1(t_0) &= AT(t_0) \\ &= A\frac{1}{4}k\alpha^2[1 + cos(2\omega t_0 - 2\delta)] \\ U_1(t_0) &= U(t_0) \\ &= \frac{1}{4}k\alpha^2[1 - cos(2\omega t_0 - 2\delta)] \end{aligned} \tag{A.37}$$

From equation A.36 we can calculate the new phase:

$$\begin{aligned} \delta_1 &= \frac{1}{2}\left[2\omega t_0 - \cos^{-1}\left(\frac{T_1 - U_1}{T_1 + U_1}\right)\right] \\ &= \frac{1}{2}\left[2\omega t_0 - \cos^{-1}\left(\frac{(A-1) + (A+1)\cos(2t_0\omega - 2\delta)}{(A+1) + (A-1)\cos(2t_0\omega - 2\delta)}\right)\right] \end{aligned} \tag{A.38}$$

The phase change is therefore:

$$\Delta\delta = \frac{1}{2}\left[2\omega t_0 - \cos^{-1}\left(\frac{(A-1) + (A+1)\cos(2t_0\omega - 2\delta)}{(A+1) + (A-1)\cos(2t_0\omega - 2\delta)}\right)\right] - \delta \tag{A.39}$$